

Preliminary Design and Control of an Operator-Assistance System Integrated into a Cobot, for Anatomical Meat-Cutting Process

Alexis Babut^a, Chedli Bouzgarrou^b, Laurent Sabourin^c and Nicolas Bouton^d

*Université Clermont Auvergne, Clermont Auvergne INP, CNRS, Institut Pascal
F-63000 Clermont-Ferrand, France*

Keywords: Operator Assistance System, Collaborative Robot, Compliant Control, Physical Human–Robot Interaction.

Abstract: This paper presents the preliminary design and control of a collaborative robotic cell for operator assistance in tasks involving soft material manipulation, such as meat cutting. The system integrates force/torque sensors and employs a Cartesian admittance controller to enable compliant, intuitive physical interaction. The mechanical design of the end-effector, the control architecture, and the communication strategy are described. Initial experiments validate the system’s ability to provide stable and responsive assistance in a physical Human–Robot Interaction (pHRI) context.

1 INTRODUCTION

While automation continues to progress across many sectors, some manual tasks remain difficult to mechanise due to their complexity, variability and reliance on human dexterity. This is especially the case in the meat industry, which faces a labour shortage and low job attractiveness. Workers are also exposed to physically demanding conditions, including repetitive movements, awkward postures and fast-paced work-flows.



Figure 1: Manual meat-cutting performed by a human operator.

Among the various tasks performed in this sector, anatomical meat cutting is particularly critical and complex. It requires fine motor skills, adaptability and force regulation, making it especially resistant to full automation. To achieve complete mechanisation, complex physical models would need to be developed and computed in real time, which remains a significant technological challenge. Consequently, the choice of an assistance approach using a collaborative robotic system is being considered.

In recent years, collaborative robots (cobots) have become essential tools in industrial and service applications requiring physical human-robot interaction (pHRI). Unlike traditional robots confined to isolated workspaces, cobots operate alongside humans, requiring advanced strategies to ensure safety, adaptability and intuitive control (Farajtabar and Charbonneau, 2024) (Ajoudani et al., 2017).

Tasks involving deformable or soft objects, such as in meat processing, pose particular challenges for pHRI due to the unpredictable nature of soft materials and variable human-applied forces. Conventional position or force control methods often fall short in achieving both compliant and robust behavior (Keemink et al., 2018).

Impedance and admittance control schemes address these challenges by regulating the dynamic interaction between motion and force. Introduced by Hogan (Hogan, 1984), impedance control is suited to motion-driven tasks, while admittance control excels when motion results from external forces, making it

^a <https://orcid.org/0009-0005-2749-1422>

^b <https://orcid.org/0000-0003-2394-1770>

^c <https://orcid.org/0000-0002-7840-9186>

^d <https://orcid.org/0000-0001-5673-5979>

especially useful for cooperative scenarios (Keemink et al., 2018) (Sharkawy and Koustoumpardis, 2022).

During meat-cutting operations, the operator typically applies significant and variable forces on the tool depending on tissue consistency and cutting direction. Accurately estimating and responding to these forces is essential for delivering effective and safe robotic assistance.

This work presents a Cartesian admittance-based assistance system integrated into a collaborative robotic cell. The system is equipped force/torque sensors for estimating user-applied efforts, enabling real-time compliant motion through a human-centred control interface.

The remainder of the paper is structured as follows: Section 2 introduces the robotic cell and tool design. Section 3 describes the control strategy and force estimation method. Section 5 presents experimental results, and Section 6 concludes with a discussion and future directions.

2 OVERVIEW OF THE ROBOTIC CELL

This section introduces the overall architecture and key components of the robotic cell. It details the robot platform, the design of the instrumented tool including its sensing capabilities, and the communication infrastructure that connects all parts of the system.

2.1 System Architecture

The robotic cell is built around a 6-axis UR30 collaborative robot (Universal Robots), mounted on a telescopic pillar (LIFTKIT-UR-620). The UR30 was selected for its high payload capacity (up to 30 kg) and compact form factor, which make it particularly suitable for force-intensive tasks such as meat cutting, where strength, precision and operator safety are essential. Its collaborative design allows safe interaction with human operators, while maintaining the flexibility required for complex trajectories in constrained environments.

The robot is controlled by a real-time Linux-based computer, and the system integrates sensing, control, and perception modules to enable accurate tracking of interactions and responsive behaviour. Communication between the robot and the control computer is established through an Ethernet cable.

The robot's end-effector is equipped with a multi-sensor instrumented tool, composed of a handle and a blade, designed for precise interaction measurement. To perceive the environment, the setup includes a 3D

vision system (StereoLabs Robotics 360 Perception Kit) and a motion tracking system (Movella MTw Awinda Research Bundle), which provide real-time scene analysis and track the human operator's pose. These perception systems are not yet employed in the current study.

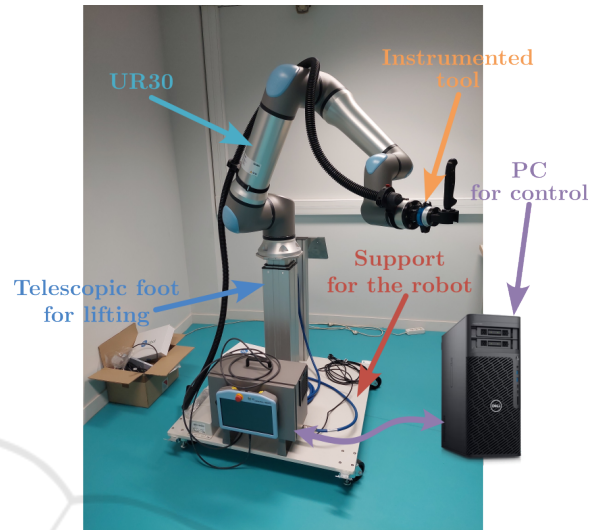


Figure 2: Overview of the robotic cell.

2.2 Design of the Instrumented Knife

This part presents the design of the instrumented tool mounted on the robot's end-effector, with a focus on its force/torque sensing capabilities and mechanical integration. The tool is intended to precisely measure interaction forces exerted by both the operator and the manipulated object.

2.2.1 Sensor Integration Prototypes

Force/torque sensors are integrated within the tool assembly to accurately capture interaction forces. The primary sensor employed is the BOTA Systems SensONE T15, selected for its high measurement precision and reliable communication performance. This sensor measures all forces applied to the mounted tool and provides better accuracy than the robot's internal force/torque sensor.

A custom mechanical adapter was designed to connect the SensONE sensor to the UR30 robot's tool flange. This adapter is manufactured using Fused Deposition Modelling (FDM) with carbon-fibre-reinforced Nylon (PA-CF), combining strength and lightweight properties.

Because two distinct forces act on the tool, the force applied by the human operator and the reaction force from the product, using only one sensor

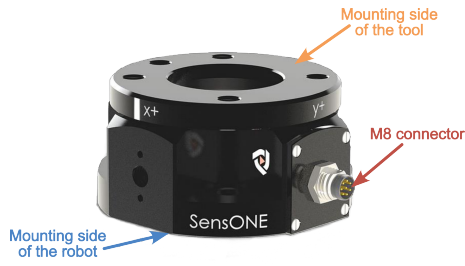
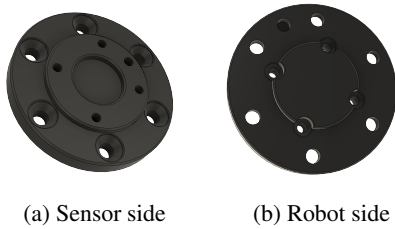


Figure 3: BOTA Systems SensONE T15 force/torque sensor.



(a) Sensor side (b) Robot side
Figure 4: SensONE to UR30 adapter.

will capture their combined effect without distinction. Therefore, a second force/torque sensor, the BOTA Systems Medusa, is required to independently measure and separate these forces.

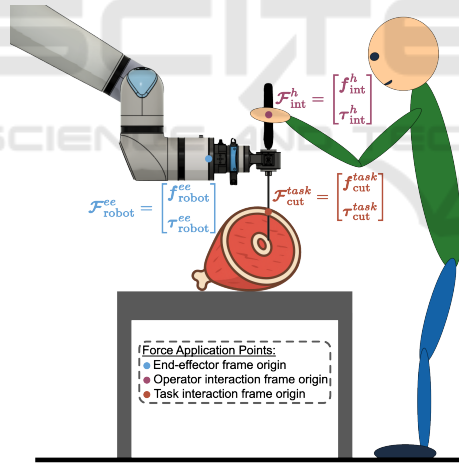


Figure 5: Forces applied on the tool during a meat-cutting task with a robot.



Figure 6: BOTA Systems Medusa force/torque sensor.

The Medusa sensor is mounted using a dedicated adapter, also printed in PA-CF using FDM. A modular design enables different assembly configurations, allowing the measurement of forces at either the handle or the blade.



(a) Top view (b) Bottom view
Figure 7: Second force/torque sensor adapter.

2.2.2 Knife Prototypes

The knife handle was modelled from a 3D scan of a real industrial knife, commonly used in meat processing. This scan served as a realistic basis for designing functional handle prototypes, which were manufactured using PLA filament through FDM (Fused Deposition Modeling) 3D printing.

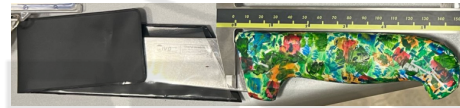


Figure 8: Industrial knife used as design reference.

To ensure mechanical robustness, the PLA handle is mounted on a welded steel substructure that serves as the structural backbone of the tool. This interface guarantees stability during high-force interactions.



(a) Steel connector (b) Assembled handle
Figure 9: Knife handle of the tool prototypes.

A push-button is integrated into the handle and connected to an analogue input on the UR30 control box. This button allows the operator to manually trigger the robot's compliant motion, offering intuitive interaction during collaborative tasks.

Finally, the complete assembly is mounted using a tool changer on the robot flange, as shown in Figure 10.

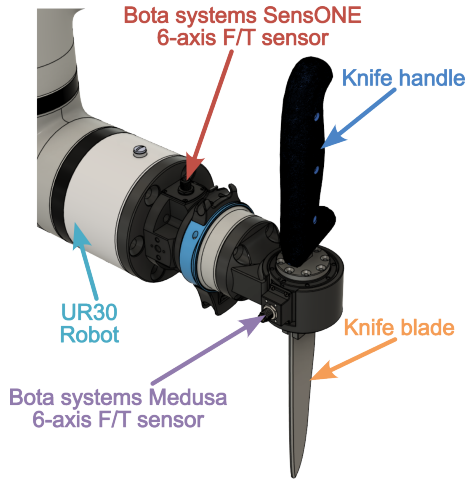


Figure 10: Tool assembly mounted on the robot tool flange.

2.3 Communication Setup

The robotic cell relies on a real-time communication architecture linking the control unit to all sensors and actuators. The UR30 robot communicates with the control computer through an Ethernet (RJ45) connection using the TCP/IP protocol. The BOTA Systems force/torque sensors, including the SensONE and Medusa models, operate over the EtherCAT protocol. They are powered using Power-over-Ethernet (PoE), and sensor data is transmitted to the computer through the LAN output of the PoE injector.

The 3D vision system is connected to the control computer through Ethernet using the TCP/IP protocol. The motion tracking system (Xsens Awinda) communicates with the computer through a dedicated USB interface using a virtual COM port (serial protocol). The tool changer is electrically linked to the robot's tool flange I/O, while the telescopic pillar is connected to the control box USB port. Both are monitored in the ROS2 environment through the `/robot_states` topic.

3 CONTROL STRATEGY

This section presents the control strategy developed for the robotic cell. Due to the limitations of the Universal Robot interface, control commands are restricted to joint positions or joint velocities. Therefore, implementing a torque-based control scheme is not possible, and the control architecture must be designed to accommodate these constraints.

The main objective is to achieve smooth and transparent interaction between the human and the robot in all directions, including rotation. The robot must re-

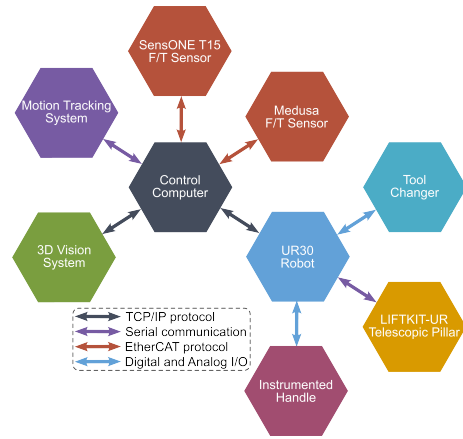


Figure 11: Communication architecture of the robotic cell.

spond naturally to the operator's guidance, allowing for intuitive and safe cooperation throughout shared tasks.

In addition to enabling compliant motion, it is necessary to regulate the cutting force applied by the tool. The control system must implement a force amplification strategy, where the cutting force in the task frame is saturated as $\mathcal{F}_{\text{cut}}^{\text{task}} = k \cdot \mathcal{F}_{\text{int}}^h$

3.1 Overview of Admittance Control

The aim of compliant control is to generate the robot's movement by emulating a mass-spring-damper system behaviour (Keemink et al., 2018) as illustrated in Figure 13.

Admittance control computes the robot's displacement $\mathbf{x}_r \in \mathbb{R}^m$ in response to external forces $\mathcal{F}_{\text{ext}} \in \mathbb{R}^m$, where $m = 6$ in our case, corresponding to Cartesian space. This relationship is modelled using a mass-spring-damper system equation:

$$\mathbf{M}_{dx}(\ddot{\mathbf{x}}_r - \ddot{\mathbf{x}}_d) + \mathbf{D}_{dx}(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}_d) + \mathbf{K}_{dx}(\mathbf{x}_r - \mathbf{x}_d) = \mathcal{F}_{\text{ext}} \quad (1)$$

Here, $\mathbf{M}_{dx}$, $\mathbf{D}_{dx}$, and $\mathbf{K}_{dx}$ are the desired Cartesian inertia, damping, and stiffness matrices respectively, while $\mathbf{x}_d$, $\dot{\mathbf{x}}_d$, and $\ddot{\mathbf{x}}_d$ denote the desired Cartesian trajectory position, velocity, and acceleration.

Using equation 1, the Cartesian acceleration resulting from the applied external forces can be expressed as:

$$\ddot{\mathbf{x}}_r = \ddot{\mathbf{x}}_d + \mathbf{M}_{dx}^{-1} \left[\mathcal{F}_{\text{ext}} - \mathbf{D}_{dx}(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}_d) - \mathbf{K}_{dx}(\mathbf{x}_r - \mathbf{x}_d) \right] \quad (2)$$

After double integration, the resulting Cartesian position error $\mathbf{e}_x = \mathbf{x}_r - \mathbf{x}_d$ is tracked by a joint-level control loop, typically handled by the robot's internal

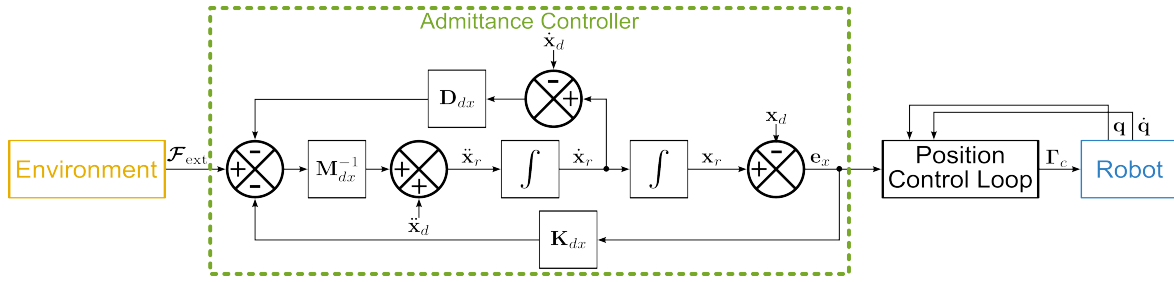


Figure 12: Block diagram of Cartesian admittance control.

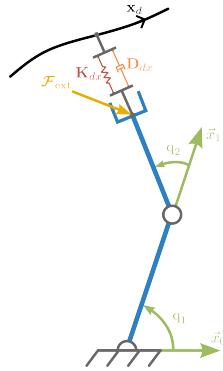


Figure 13: Schematic of the robot's compliant model principle.

controller. If the robot's dynamics are known, an inverse dynamics model combined with a feedback controller (such as PID or PD) can be used to regulate the joint motion and follow the reference trajectory. The resulting control torques Γ_c are applied to the robot's actuators, as illustrated in Figure 12.

3.2 System Coordinate Frames

Accurate control and interpretation of sensor data require a defined set of coordinate frames for the robot, the tool, and the associated sensors. These frames serve as spatial references for motion planning, force estimation, and control execution. Figure 14 shows an overview of the coordinate frames of the robotic system, including the robot base frame, individual joint frames, and the end-effector frame.

3.3 Estimation of Human-Applied Forces

To estimate the forces applied by the human operator on the handle, a six-axis force/torque sensor (Medusa) is mounted directly on the handle. This sensor measures the interaction forces transmitted through the handle. The method presented here is used to identify the sensor bias and recover the true force exerted by the operator. The same procedure is applied to

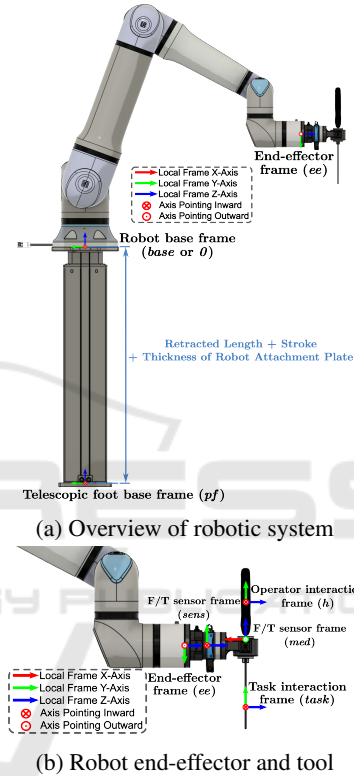


Figure 14: Coordinate frames defined for the Robotic system.

the primary force/torque sensor (SensONE), which is mounted on the robot tool flange thanks to an adapter piece, to estimate external forces acting on the tool.

3.3.1 Bias and Handle Parameter Identification

Force/torque sensors typically have intrinsic bias, causing non-zero output even when no external load is applied. This bias can result from installation preloads, component weight, or dynamic effects during motion.

Under static conditions without external contact forces, the sensor output mainly reflects the bias and the gravitational force due to the handle's mass. The handle's mass m_{ha} is measured beforehand using a

precision scale and then introduced in the identification process. The sensor output is modelled as:

$$\begin{aligned} \mathbf{f}_{\text{measured}}^{\text{med}} &= \mathbf{f}_{\text{bias}}^{\text{med}} + m_{\text{ha}} \mathbf{g}^{\text{med}} \\ \boldsymbol{\tau}_{\text{measured}}^{\text{med}} &= \boldsymbol{\tau}_{\text{bias}}^{\text{med}} - m_{\text{ha}} {}^{\text{med}}\mathbf{P}_{\text{CoM}} \times \mathbf{g}^{\text{med}} \end{aligned} \quad (3)$$

Here, $\mathbf{f}_{\text{bias}}^{\text{med}}$ and $\boldsymbol{\tau}_{\text{bias}}^{\text{med}}$ denote the constant force and torque sensor biases, m_{ha} is the mass of the handle, and ${}^{\text{med}}\mathbf{P}_{\text{CoM}}$ is the position vector from the sensor origin to the handle's centre of mass, expressed in the sensor frame, as illustrated in Figure 15. The vector $\mathbf{g}^{\text{med}}$ represents the gravity vector expressed in the sensor frame, computed from the gravity vector in the robot base frame:

$$\mathbf{g}^{\text{base}} = [0 \quad 0 \quad -9.81]^T \text{ m/s}^2$$

Given ${}^{ee}\mathbf{A}_{\text{med}}$ as the rotation matrix of the sensor relatively to the end-effector frame, and ${}^{\text{base}}\mathbf{A}_{ee}$ as the end-effector orientation in the base frame, the gravity vector in the sensor frame is given by:

$$\mathbf{g}^{\text{med}} = \left({}^{\text{base}}\mathbf{A}_{ee} {}^{ee}\mathbf{A}_{\text{med}} \right)^T \mathbf{g}^{\text{base}} \quad (4)$$

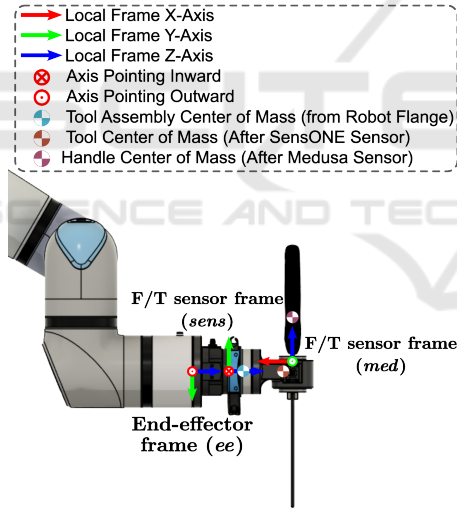


Figure 15: Sensor frames and tool center of mass.

For each static pose i , the measurements satisfy the linear system:

$$\underbrace{\begin{bmatrix} \mathbf{f}_1^{\text{med}} - m_{\text{ha}} \mathbf{g}_1^{\text{med}} \\ \boldsymbol{\tau}_1^{\text{med}} \\ \vdots \\ \mathbf{f}_N^{\text{med}} - m_{\text{ha}} \mathbf{g}_N^{\text{med}} \\ \boldsymbol{\tau}_N^{\text{med}} \end{bmatrix}}_{\mathbf{b} \in \mathbb{R}^{6N}} = \underbrace{\begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & -m_{\text{ha}} [\mathbf{g}_1^{\text{med}}]_{\times} \\ \vdots & \vdots & \vdots \\ \mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & -m_{\text{ha}} [\mathbf{g}_N^{\text{med}}]_{\times} \end{bmatrix}}_{\mathbf{A} \in \mathbb{R}^{6N \times 9}} \underbrace{\begin{bmatrix} \mathbf{f}_{\text{bias}}^{\text{med}} \\ \boldsymbol{\tau}_{\text{bias}}^{\text{med}} \\ {}^{\text{med}}\mathbf{P}_{\text{CoM}} \end{bmatrix}}_{\mathbf{X} \in \mathbb{R}^9} \quad (5)$$

Here, $[\mathbf{g}_i^{\text{med}}]_{\times}$ is the skew-symmetric matrix associated with the gravity vector at pose i , in the sensor frame. Using measurements from N static poses, this overdetermined system is solved thanks to least squares to identify unknown parameters:

$$\mathbf{X} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (6)$$

3.3.2 Gravity Compensation

After estimating bias, handle mass, and centre of mass, sensor outputs are compensated to isolate the human-applied forces.

The sensor orientation relative to the base frame is ${}^{\text{base}}\mathbf{A}_{\text{med}}$, and the gravity vector in the sensor frame is:

$$\mathbf{g}^{\text{med}} = \left({}^{\text{base}}\mathbf{A}_{\text{med}} \right)^T \mathbf{g}^{\text{base}} \quad (7)$$

The gravitational force and moment on the sensor are:

$$\begin{aligned} \mathbf{f}_{\text{gravity}}^{\text{med}} &= m_{\text{ha}} \mathbf{g}^{\text{med}} \\ \boldsymbol{\tau}_{\text{gravity}}^{\text{med}} &= {}^{\text{med}}\mathbf{P}_{\text{CoM}} \times \mathbf{f}_{\text{gravity}}^{\text{med}} \end{aligned} \quad (8)$$

The compensated force and moment in the sensor frame are then:

$$\begin{aligned} \mathbf{f}_{\text{ext}}^{\text{med}} &= \mathbf{f}_{\text{measured}}^{\text{med}} - \mathbf{f}_{\text{bias}}^{\text{med}} - \mathbf{f}_{\text{gravity}}^{\text{med}} \\ \boldsymbol{\tau}_{\text{ext}}^{\text{med}} &= \boldsymbol{\tau}_{\text{measured}}^{\text{med}} - \boldsymbol{\tau}_{\text{bias}}^{\text{med}} - \boldsymbol{\tau}_{\text{gravity}}^{\text{med}} \end{aligned} \quad (9)$$

The resulting vector $\mathcal{F}_{\text{ext}}^{\text{med}} = \left[\left(\mathbf{f}_{\text{ext}}^{\text{med}} \right)^T \quad \left(\boldsymbol{\tau}_{\text{ext}}^{\text{med}} \right)^T \right]^T \in \mathbb{R}^6$ represents the force and torque applied by the human operator, expressed in the sensor frame and compensated for sensor bias and gravitational effects.

3.4 Compliant Control

Once the force applied by the human operator has been estimated according to Equation 9, it is first expressed in the operator frame h (see Equation 10), which is attached to the handle and defines the human-tool interaction interface (see Figure 14).

$$\mathcal{F}_{\text{int}}^h = \begin{bmatrix} \mathbf{f}_{\text{int}}^h \\ \boldsymbol{\tau}_{\text{int}}^h \end{bmatrix} = \begin{bmatrix} {}^{\text{med}}\mathbf{A}_h & \mathbf{0}_{3 \times 3} \\ [{}^{\text{med}}\mathbf{P}_h]_{\times} & {}^{\text{med}}\mathbf{A}_h \end{bmatrix} \mathcal{F}_{\text{ext}}^{\text{med}} \quad (10)$$

Here, ${}^{\text{med}}\mathbf{A}_h$ denotes the rotation matrix of the interaction frame h to the sensor frame (med), and ${}^{\text{med}}\mathbf{P}_h$ is the position vector of the origin of frame h expressed in the sensor frame.

Rather than using on position control, the control strategy presented in this section is based on velocity control, which is more appropriate for the dynamic characteristics of the intended task. In the same way,

an admittance-based control law is used to compute the Cartesian velocity in response to external forces applied by the human operator.

The desired Cartesian velocity is obtained by integrating the acceleration generated by a mass–spring–damper model, as defined in Equation 2. This velocity determines how the robot should move within the task space in order to respond compliantly to the operator’s efforts. It is initially computed in the operator frame h and then transformed into the robot end-effector frame, as shown in Equation 11.

$$\dot{\mathbf{x}}_r^{ee} = \begin{bmatrix} \mathbf{v}_r^{ee} \\ \boldsymbol{\omega}_r^{ee} \end{bmatrix} = \underbrace{\begin{bmatrix} {}^{ee}\mathbf{A}_h & -{}^{ee}\mathbf{A}_h [{}^h\mathbf{P}_{ee}]_{\times} \\ \mathbf{0}_{3 \times 3} & {}^{ee}\mathbf{A}_h \end{bmatrix}}_{\mathbf{Ad}^{ee}\mathbf{T}_h \in \mathbb{R}^{6 \times 6}} \dot{\mathbf{x}}_r^h \quad (11)$$

Here, $\dot{\mathbf{x}}_r^h = \left[(\mathbf{v}_r^h)^T \quad (\boldsymbol{\omega}_r^h)^T \right]^T$ denotes the compliant motion resulting from the control law, expressed in the interaction (operator) frame h . The rotation matrix ${}^{ee}\mathbf{A}_h$ represents the orientation of frame h with respect to the robot end-effector frame. The vector ${}^h\mathbf{P}_{ee}$ represents the position of the end-effector origin expressed in frame h , and $[{}^h\mathbf{P}_{ee}]_{\times}$ denotes its skew-symmetric matrix. The full matrix multiplying $\dot{\mathbf{x}}_r^h$ corresponds to the adjoint transformation $\mathbf{Ad}^{ee}\mathbf{T}_h$, which maps spatial velocity (twists) from the interaction frame to the robot end-effector frame.

The resulting Cartesian velocity, now expressed in the end-effector frame, is mapped to joint velocities $\dot{\mathbf{q}}_c$ using the inverse of the robot Jacobian ${}^0\mathbf{J}_{ee}^{-1}$. These joint velocity commands are subsequently sent to the low-level joint velocity controller of the robot for execution.

3.4.1 Example of a Compliant Motion

As an initial implementation, a simple compliant control scheme was developed based on the general form of the admittance equation, given in Equation 12:

$$\ddot{\mathbf{x}}_r^h = \mathbf{M}_{dx}^{-1} \left(\mathcal{F}_{\text{int}}^h - \mathbf{D}_{dx} \dot{\mathbf{x}}_r^h \right) \quad (12)$$

In this formulation, $\mathbf{M}_{dx}$ and $\mathbf{D}_{dx}$ denote the desired Cartesian inertia and damping matrices, respectively. The variables $\ddot{\mathbf{x}}_r^h$ and $\dot{\mathbf{x}}_r^h$ represent the resulting Cartesian acceleration and velocity in the interaction frame h .

In this case, the stiffness matrix $\mathbf{K}_{dx}$ is set to the zero matrix $\mathbf{0}_{6 \times 6}$, which means that the stiffness effect typically present in admittance control is removed. Consequently, the robot motion corresponds to displacement induced purely by the operator’s interaction force, without any restoring force pulling the robot back towards a desired or initial position $\mathbf{x}_d$.

The control parameters used in this implementation are as follows:

$$\mathbf{M}_{dx} = \begin{bmatrix} 15.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 15.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.4 \end{bmatrix}$$

$$\mathbf{D}_{dx} = \begin{bmatrix} 30.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 30.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 30.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8 \end{bmatrix}$$

4 ROS2 ENVIRONMENT

The control environment was implemented in ROS2 and configured with a single force/torque sensor (Medusa). The UR ROS2 driver package and all related nodes operated at 500 Hz, matching the robot’s internal control loop. The dedicated node for the Medusa sensor ran at 800 Hz, with measurements interpolated in the compliant control node to align with the robot control frequency.

The system architecture, shown in Figure 17, included three main ROS2 nodes. The *F/T Sensor Medusa* node published sensor data, filtered internally by the sensor with a built-in cutoff frequency, to the compliant control node, while also interfacing with the EtherCAT reader Python script through an internal TCP socket. The *Compliant Control* node subscribed to the sensor data and published the end-effector target velocity. Finally, the *Cartesian Velocity Controller* node computed joint velocity commands, which were transmitted to the UR ROS2 driver for execution by the UR30 robot.

System-level optimisation included CPU isolation and real-time scheduling in FIFO mode. Dynamic memory allocation was reduced and computational cost minimised, providing more deterministic behaviour compared with standard implementations.

Sensor initialisation and setup were carried out through dedicated safety routines, triggered manually from the terminal. In this configuration, only the built-in safety functions of the UR30 robot were enabled. Future extensions will integrate external safety layers, including 3D vision. The Cartesian velocity controller also accounted for the robot’s maximum joint velocity and torque limits, saturating commands when necessary to prevent unsafe execution.

Jitter, which quantifies the variability of the control loop period, can be expressed as the difference

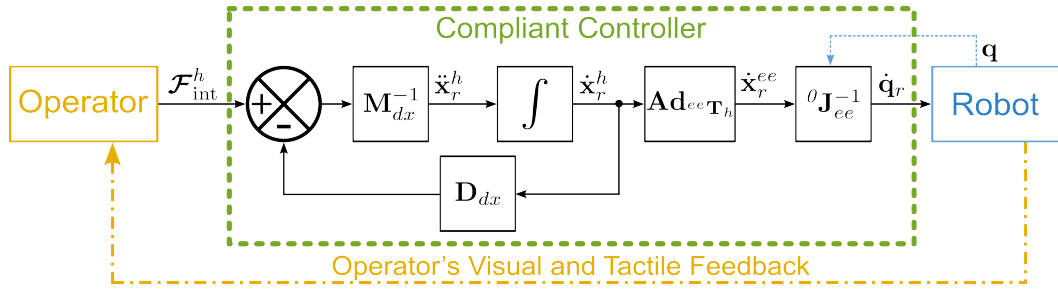


Figure 16: Block diagram of the implemented compliant control.

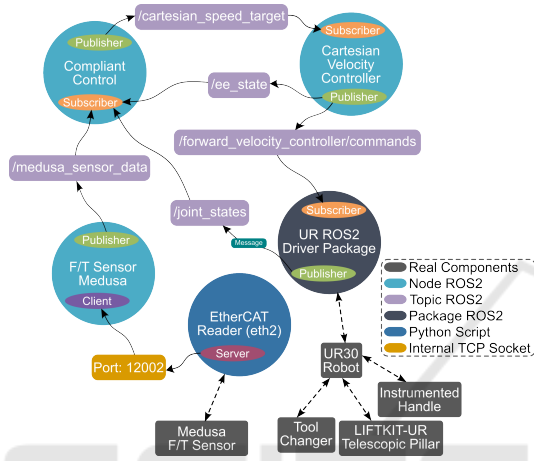


Figure 17: ROS2 control environment with one force/torque sensor.

between the current loop period T_{cur} and the nominal period T_{nom} :

$$\text{Jitter} = |T_{\text{cur}} - T_{\text{nom}}| \quad (13)$$

Figures 18a and 18b show the measured loop jitter for the compliant control node without and with a real-time (RT) kernel, respectively. Using the standard kernel, significant jitter peaks exceeding 500 μs were observed, whereas the RT kernel maintained jitter consistently below 250 μs . This demonstrates a clear improvement in temporal determinism when using real-time scheduling.

Figures 19 shows the end-to-end latency of the control loop. Without real-time kernel, the mean latency was 1881.4 μs with peaks above 2 ms, indicating delays within the 500 Hz loop. With real-time kernel, the mean latency decreased to 541.2 μs , with no peaks beyond 2 ms.

5 EXPERIMENTAL RESULTS

The previous sections described the integration of a dual force/torque sensing architecture into the instrumented tool. However, the preliminary experiments

reported here were conducted using only the Medusa sensor mounted on the tool.

The test scenario consisted of standard cutting movements representative of a meat processing task. Specifically, slow forward motions were executed along the cutting direction, followed by rapid return movements towards another part of the meat piece. These simplified yet representative conditions were chosen to evaluate the dynamic behaviour of the admittance-based compliant controller under human interaction in a meat-cutting use case. In this configuration, only the displacement of the robot was controlled in response to the operator's applied force, as amplification features were implemented but not tested due to the use of a single sensor.

Figure 21a shows the estimated external force applied by the human operator, reconstructed from the Medusa sensor data after bias and gravity compensation. In response, the compliant controller generates a Cartesian velocity command as illustrated in Figure 21b.

To evaluate how well the robot tracks this reference motion, Figure 23a compares the commanded Cartesian velocity with the measured velocity, derived from joint encoders using the forward velocity kinematics:

$$\dot{\mathbf{x}}_{\text{robot}}^{\text{ee}} = {}^0\mathbf{J}_{\text{ee}} \dot{\mathbf{q}}$$

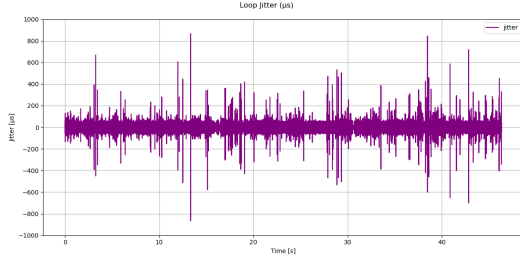
The correlation between both signals confirms the robot's internal controller accurately follows the velocity commands.

Figure 23b presents the estimated force applied by the robot at the end-effector. This is obtained from the joint torques thanks to the relation:

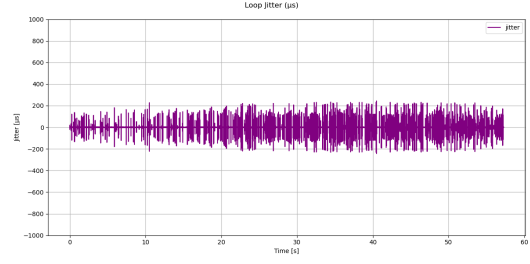
$$\mathcal{F}_{\text{robot}}^{\text{ee}} = {}^0\mathbf{J}_{\text{ee}}^{-T} \boldsymbol{\Gamma}$$

Excluding contributions from the tool's own dynamics and gravity. The plot highlights the robot's passive response to external forces applied by the human operator.

The transparency of the compliant control was evaluated using the variable Y_{diff} , defined as the absolute difference between the measured and desired

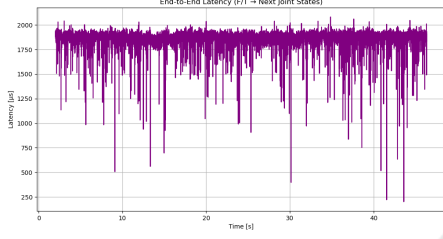


(a) Loop jitter without a real-time kernel.

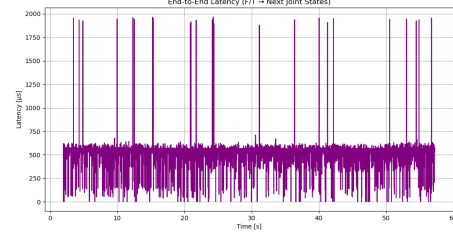


(b) Loop jitter with a real-time kernel.

Figure 18: Comparison of loop jitter for the compliant control node under standard and real-time kernels.



(a) End-to-end latency without a real-time kernel.



(b) End-to-end latency with a real-time kernel.

Figure 19: End-to-end latency from sensor data acquisition to robot command execution.

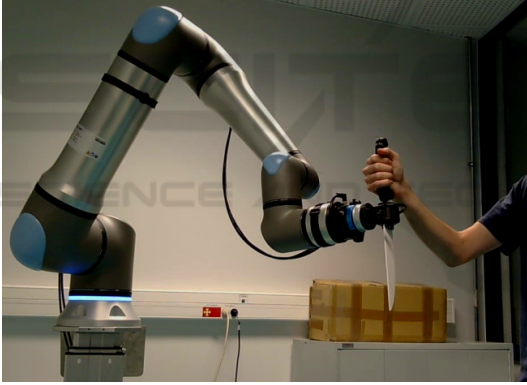


Figure 20: Illustration of the human operator applying a force on the handle, resulting in a tool motion.

admittance gains:

$$Y_d = \frac{\dot{x}_r^h}{\mathcal{F}_{int}^h}, \quad Y_{mes} = \frac{\dot{x}_{mes}^h}{\mathcal{F}_{int}^h}, \quad Y_{diff} = |Y_{mes} - Y_d| \quad (14)$$

where $\dot{x}_r^h$ is the commanded end-effector velocity in the human reference frame, $\dot{x}_{mes}^h$ is the measured end-effector velocity in the same frame, and $\mathcal{F}_{int}^h$ is the force applied by the human operator.

This first indicator of transparency for the compliant motion demonstrates good performance during standard movements: the robot successfully follows the human-induced motion, particularly during phases of nearly constant velocity. However, when rapid and large forces are applied (notably around 40s in the plots), the co-movements are insufficiently

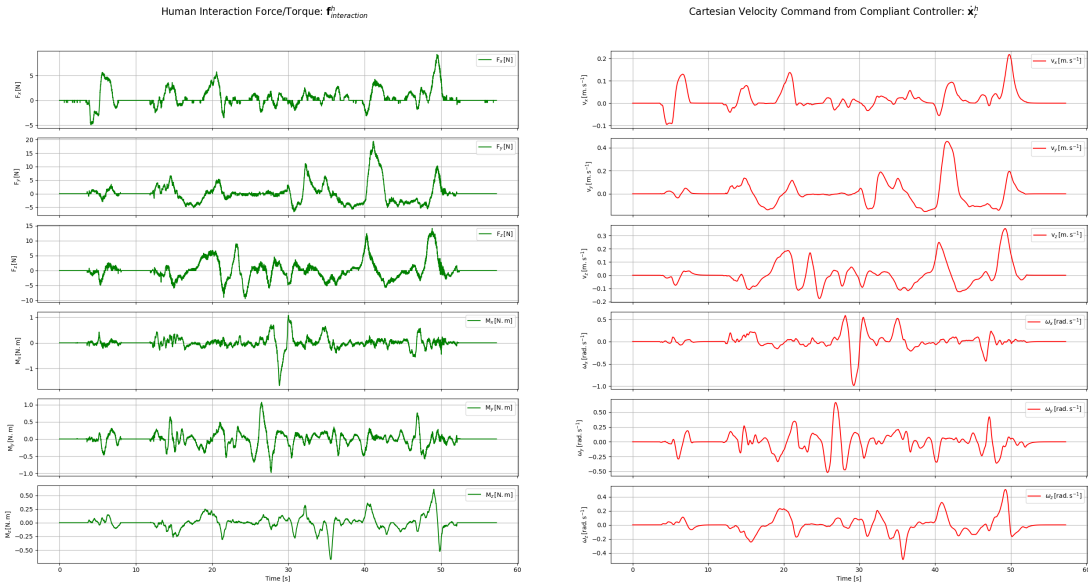
smooth. The motion is strongly damped, which results in peaks in Y_{diff} . This limitation is primarily due to the damping matrix used in the current implementation, which reduces the operator's intended dynamics. A second contributing factor could be the robot itself, as observed in Figure 23a, where a significant discrepancy exists between the desired end-effector velocity from the compliant controller and the measured value.

Future transparency indicators will be investigated, focusing in particular on the frequency response of the admittance rather than a simple absolute deviation.

A promising direction for improvement is the implementation of a variable damping matrix, as proposed in (Sharkawy and Koustoumpardis, 2022), to better reflect the operator's intended accelerations. This is particularly relevant for anatomical tasks such as meat cutting, where the human performs motions with frequent changes in direction and acceleration amplitude.

These results are consistent with the existing literature and reinforce the practicality of implementing Cartesian admittance control by generating joint velocity commands. This approach benefits from low-level integration, allowing the robot's internal controller to ensure stable and responsive behaviour through the management of filtering, actuation limits and dynamic consistency.

However, several limitations must be taken into



(a) Estimated human–robot interaction force.

(b) Resulting Cartesian velocity.

Figure 21: Input and output of the compliant controller (expressed in the interaction frame h).

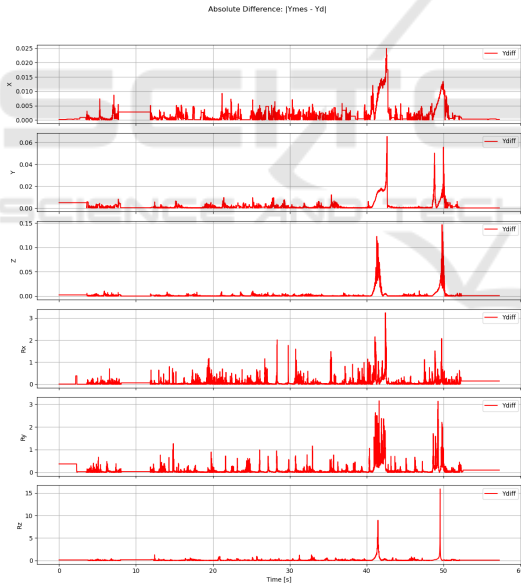


Figure 22: Absolute difference between measured and desired admittance gains, Y_{diff} , during compliant control.

account. Firstly, mapping Cartesian velocities to joint velocities using the Jacobian can become ill-conditioned near singularities, resulting in large joint movements or noisy behaviour. Secondly, this strategy depends on accurate dynamic modelling and sensor calibration, as any error in force estimation directly affects motion quality. Additionally, when the control loop does not operate in real time or lacks suf-

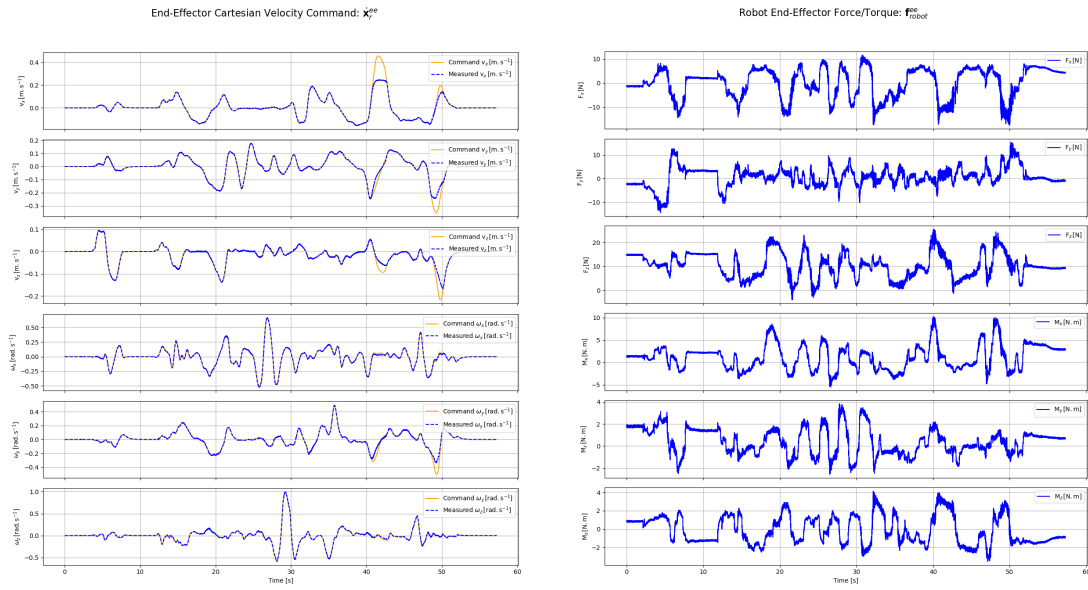
ficient frequency, latency can reduce responsiveness and compromise safety.

6 CONCLUSION

This paper has presented the design and preliminary results of the control of an operator-assistance system integrated into a collaborative robot for the manipulation of soft bodies, such as in meat-cutting tasks. The system combines a carefully instrumented end-effector with a Cartesian admittance controller to enable compliant and intuitive physical human-robot interaction.

The mechanical design of the instrumented tool, featuring dual force/torque sensors, has been validated to effectively distinguish between operator-applied forces and interaction forces with the soft object. The communication infrastructure and control strategy, based on velocity commands and admittance control, have demonstrated promising preliminary results in delivering responsive and stable assistance.

Future work will focus on in-depth experimental validation in realistic industrial scenarios, on improving the robot's force control capabilities using a method distinct from that proposed in (Hamad et al., 2021), and on integrating adaptive control schemes to manage the variability in the direction of force and acceleration typically applied by the operator in tasks such as meat cutting. Moreover, the integration of advanced perception modalities and operator intent



(a) Comparison between commanded and measured end-effector velocity.

(b) Estimated force applied by the robot at the end-effector.

Figure 23: Robot behaviour during compliant control.

recognition is expected to further enhance the system's robustness and usability.

Overall, this study lays the groundwork for the development of collaborative robotic systems capable of safely and efficiently manipulating deformable objects, while providing ergonomic assistance in tasks involving soft body interaction. This opens new opportunities for automation in complex, human-centred industrial environments.

ACKNOWLEDGEMENTS

This work is funded by Interreg Sudoe, a European Union funding program to support regional development and cohesion in the regions of southwestern Europe, as part of the ROBOTA Sudoe project (S1/1.1/P0125): Robotics, Automation, and Digitization as Drivers of Competitiveness and Growth for SMEs.

REFERENCES

- Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., and Khatib, O. (2017). Progress and prospects of the human-robot collaboration. *Autonomous Robots*, 42(5):957–975.
- Farajtabar, M. and Charbonneau, M. (2024). The path

towards contact-based physical human-robot interaction. *Robotics and Autonomous Systems*, 182:104829.

- Hamad, Y. M., Aydin, Y., and Basdogan, C. (2021). Adaptive human force scaling via admittance control for physical human-robot interaction. *IEEE Transactions on Haptics*, 14(4):750–761.

- Hogan, N. (1984). Impedance control: An approach to manipulation. In *1984 American Control Conference*. IEEE.

- Keemink, A. Q., van der Kooij, H., and Stienen, A. H. (2018). Admittance control for physical human-robot interaction. *The International Journal of Robotics Research*, 37(11):1421–1444.

- Sharkawy, A.-N. and Koustoumpardis, P. N. (2022). Human-robot interaction: A review and analysis on variable admittance control, safety, and perspectives. *Machines*, 10(7):591.