2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Tactile Memory for Continuous Policy Blending in Unified Force-Impedance Control

Kübra Karacan[1], Ayça Demir[1], Doğukan Tosun[1], Feyza Nur Söğüt[1], Robin Jeanne Kirschner[1,2], Hamid Sadeghian[1], and Sami Haddadin[3]

***Abstract*— As of today, automating contact-rich industrial manipulation processes, such as insertion, plugging, and screw-driving, is tedious and requires expert knowledge. The processes consist of programmable, common action units, like moving to a pose and establishing contact. However, the user still has to decide on fixed transition conditions to successfully complete each sub-action. Instead, we introduce a tactile memory-driven policy blending framework based on unified force-impedance control to enable autonomous transitions. At the core of our approach lies a structured representation of manipulation as a sequence of basic operations combined into relevant processes, each governed by real-time sensory feedback and annotated with process quality metrics (PQMs) that capture motion, force, and energy-level interactions. A bidirectional long-short-term memory (BiLSTM) model encodes recent PQM histories to determine the success of basic operations. Later, soft blending weights are generated, allowing smooth, adaptive transitions between operations without manual phase definition. To ensure functional safety during contact, we integrate an energy tank mechanism that enforces passivity by regulating energy exchange. The resulting control scheme enables robust and continuous tactile manipulation across variations in object geometry and spatial configurations. Experimental validation across four processes, five objects, and two position variants demonstrates successful transfer and resilience to position disturbances. Our findings highlight that learned tactile memory and quality feedback embedded in the control loop provide a principled foundation for intelligent, transferable manipulation, enabling fully autonomous process planning and execution in the future.**

I. Introduction

Industrial robots are typically programmed using predefined sequences of basic operations (BOs), structured through switching strategies with fixed state or time-based transitions [1]. While effective for structured processes like pick-and-place, these approaches struggle in contact-rich and multi-modal processes such as insertion, plugging, and screw-driving, where adaptation is essential [2]–[4].

Humans, by nature, can regulate their actions based on sensory feedback, memory, and an implicit, learned understanding of the progression of processes. For instance, when inserting a key into the door lock, we may wiggle its tip even before touching the lock to open the door quickly. This anticipatory, feedback-driven blending, absent in most robotic systems, is the foundation of robust manipulation in contact-rich settings.

The authors are with [1] the Chair of Robotics and Systems Intelligence, MIRMI-Munich Institute of Robotics and Machine Intelligence, Technical University of Munich (TUM), Germany, [2] TUM Hospital, Department for Orthopedics and Sport Orthopedics, and [3] Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, United Arab Emirates. kuebra.karacan@tum.de

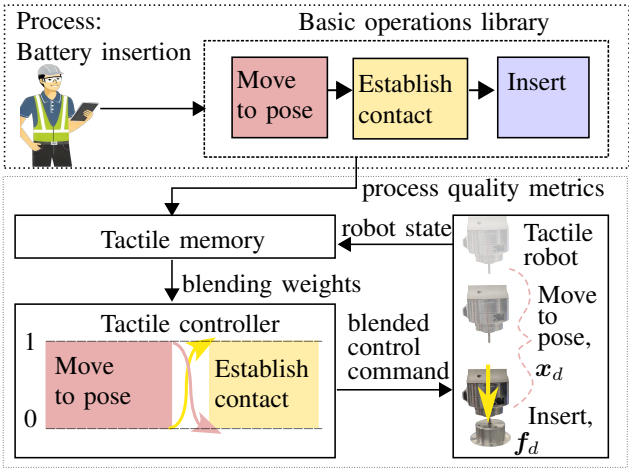

Fig. 1: **Tactile Memory for Continuous Policy Blending.** A human expert selects a process decomposed into reusable basic operations (BOs). The tactile memory determines the success of the current BO by comparing the history of process quality metrics (PQMs) with the current sensory data, producing real-time blending weights that allow continuous transitions between BOs within a unified force-impedance control framework.

To move beyond rigid control schemes, robots must integrate process memory, real-time feedback, and phase-awareness into their execution [5]. Ideally, a human should specify a high-level process, e.g., "insert peg with low tolerance". At the same time, the robot intelligently blends relevant BOs, modulating force and motion in response to evolving process dynamics without having to finetune any switching condition.

To address these gaps, we propose a pipeline for *tactile memory for continuous policy blending in unified force-impedance control* that enables smooth, feedback-driven transitions between BOs at the control level, as outlined in Fig. 1. Unlike systems that rely on discrete cues to trigger transitions, our approach embeds real-time process understanding into the control loop via 32 process quality metrics (PQMs), capturing position, velocity, force, distance-to-goal error, kinetic and potential energies. A bidirectional Long Short-Term Memory (BiLSTM) encodes recent PQMs' histories to predict BO success, and soft blending weights are subsequently generated via a sigmoid function, enabling smooth, adaptive transitions between operations, which are then fed to unified force-impedance control (UFIC) to compute a blended control wrench. We incorporate an energy tank mechanism that enforces passivity by regulating power flow to ensure functional safety, especially during contact-rich interactions.

979-8-3315-8160-2/26/$31.00 ©2026 IEEE

By combining low-level force-motion control with high-level memory and feedback integration, the magic of using AI to refine traditionally controlled and modeled interactions, our method enables robots to transfer robustly and securely perform complex manipulation processes, without relying on hard-coded state transitions or manually tuned thresholds.

A. Problem Statement

We investigate whether a robot can leverage past successful executions to learn a transferable and real-time control blending strategy that enables context-aware, tactile memory-informed continuous transitions across unseen instances of contact-rich processes.

B. Related Work

Contact-rich industrial processes require precise coordination between motion and force. Finite state machines dominate classical implementations [6], offering structured yet rigid execution.

Force control remains central to these processes, but typical methods use fixed setpoints or hand-tuned impedance gains [7]–[11]. Recent work explores adaptive impedance or learning-based approaches to improve contact robustness [12]–[18], but most still rely on discrete switching or external supervision.

Extensions using Dynamic Motion Primitives (DMPs) [19] and phase-based modulation [20] improve generalization but remain limited by fixed-phase assumptions and limited reactivity.

The FORGE framework [21] highlights the potential of modular policies and learned transitions, but its reliance on reinforcement learning and phase segmentation introduces design complexity and brittle handovers. Process quality metrics (PQMs), used in recent works for evaluation or gain tuning [5], [22]–[24], have not yet been tightly integrated into the control loop.

In contrast, our framework uses 32 carefully selected PQMs as first-class feedback signals for real-time control blending. We unify perception, control, and safety under one architecture by embedding temporal metric sequences into a recurrent policy.

C. Contributions

This paper introduces a novel control architecture for intelligent, adaptive robotic manipulation through tactile memory-driven blending. In this work, we present

I. **Tactile memory for continuous policy blending:** A BiLSTM encodes the carefully chosen 32 PQMs' histories to decide success, and soft blending weights are subsequently generated via a sigmoid function, enabling smooth and adaptive transitions between operations.
II. **Unified control architecture:** We embed this policy blending into a force-impedance controller for simultaneous motion and force tracking.
III. **Passivity-based safety:** An energy tank mechanism enforces safe power exchange and ensures passivity during contact.

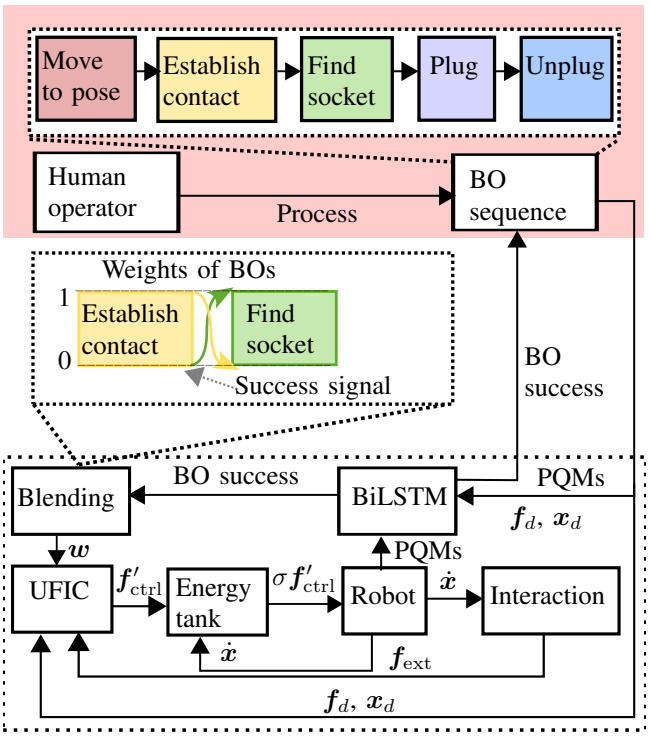

Fig. 2: **Pipeline of Tactile Memory for Continuous Policy Blending in Unified Force-Impedance Control.** The human operator specifies the target process, which is mapped to a sequence of basic operations (BOs). The robot collects real-time process quality metrics (PQMs) during execution and feeds them into a BiLSTM network that predicts BO success. The resulting blending weights are used to interpolate control wrenches between operations. The unified force-impedance controller (UFIC) generates physically plausible interaction behavior, while an energy tank regulates power injection and enforces passivity for functional safety.

IV. **Experimental validation:** We evaluate our method on four processes, peg insertion, USB plugging-unplugging, and screw-driving, demonstrating successful transferability across five objects and two pose variations.

II. Methodology

We propose a tactile memory for a continuous policy blending framework that enables the safe, continuous execution of contact-rich manipulation processes in industrial robotics. This approach overcomes the limitations of traditional state-machine-based schemes, which typically rely on fixed-phase transitions and limited feedback integration. Instead, our method blends predefined physical primitives based on real-time sensory interpretation and energy-aware regulation, as shown in Fig. 2. The framework comprises four key modules: i.) a library of basic operations (BOs), ii.) unified force-impedance control, iii.) a BiLSTM-based BO success decision and smooth policy blending strategy, and iv.) an energy tank mechanism for passivity enforcement.

A. Process and Basic Operations Library

We begin by organizing each manipulation process into a sequence of functionally grounded phases

1 - Free-space motion to an approximate target location,
2 - Contact establishment with the environment,

3 - Feature localization, such as detecting a hole, screw, or socket,
4 - The main manipulation action, like insertion, plugging, unplugging, or screw-driving.

To represent these phases, we define a modular set of BOs. Each BO produces a control wrench $\boldsymbol{f}_{\text{ctrl}} \in \mathbb{R}^6$ and is parameterized by a desired Cartesian pose $\boldsymbol{x}_{\text{d}} \in \mathbb{R}^6$ and force $\boldsymbol{f}_{\text{d}} \in \mathbb{R}^6$. The BOs include: move-to-pose, establish contact, find hole-socket-screw, insert, plug, unplug, and screw. These BOs can be reused across different processes and support both learning, capability transfer, and runtime adaptation.

B. Unified Force-Impedance Control

Each BO uses a Cartesian-space control formulation that unifies impedance and force control. The Cartesian dynamics of a torque-controlled robot are given by

$$\boldsymbol{M}_{\text{C}}(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{C}_{\text{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{x}} + \boldsymbol{f}_{\text{g}} = \boldsymbol{f}_{\text{robot}} + \boldsymbol{f}_{\text{ext}}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^6$ is the Cartesian pose, $\boldsymbol{M}_{\text{C}} \in \mathbb{R}^{6\times6}$ the Cartesian mass matrix, $\boldsymbol{C}_{\text{C}} \in \mathbb{R}^{6\times6}$ the Coriolis matrix, $\boldsymbol{f}_{\text{g}} \in \mathbb{R}^6$ the gravity wrench, and $\boldsymbol{f}_{\text{ext}} \in \mathbb{R}^6$ the external wrench. The total robot wrench $\boldsymbol{f}_{\text{robot}} \in \mathbb{R}^6$ is

$$\boldsymbol{f}_{\text{robot}} = \boldsymbol{f}_{\text{ctrl}} + \boldsymbol{f}_{\text{g}}\,, \tag{2}$$

where $\boldsymbol{f}_{\text{ctrl}}$ is the sum of impedance $\boldsymbol{f}_{\text{imp}} \in \mathbb{R}^6$ and force control $\boldsymbol{f}_{\text{frc}} \in \mathbb{R}^6$ terms

$$\boldsymbol{f}_{\text{ctrl}} = \boldsymbol{f}_{\text{imp}} + \boldsymbol{f}_{\text{frc}}\,. \tag{3}$$

The impedance term $\boldsymbol{f}_{\text{imp}}$ regulates motion through mass-spring-damper dynamics [25]

$$\boldsymbol{f}_{\text{imp}} = \boldsymbol{K}_{\text{C}}\tilde{\boldsymbol{x}} + \boldsymbol{D}_{\text{C}}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{M}_{\text{C}}\ddot{\boldsymbol{x}}_{\text{d}} + \boldsymbol{C}_{\text{C}}\dot{\boldsymbol{x}}_{\text{d}}\,, \tag{4}$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{x}_{\text{d}} - \boldsymbol{x} \in \mathbb{R}^6$, $\boldsymbol{K}_{\text{C}}, \boldsymbol{D}_{\text{C}} \in \mathbb{R}^{6\times6}$ are the stiffness and damping matrices, and $\boldsymbol{x}_{\text{d}}, \dot{\boldsymbol{x}}_{\text{d}}, \ddot{\boldsymbol{x}}_{\text{d}} \in \mathbb{R}^6$ denote the desired pose, velocity, and acceleration.

The force control term $\boldsymbol{f}_{\text{frc}}$ compensates for force errors while tracking the desired feed-forward force reference $\boldsymbol{f}_{\text{d}}$ [26]. It is computed as

$$\boldsymbol{f}_{\text{frc}} = \boldsymbol{f}_{\text{d}} + \boldsymbol{K}_{\text{p}}\tilde{\boldsymbol{f}}_{\text{ext}} + \boldsymbol{K}_{\text{i}}\int_0^t \tilde{\boldsymbol{f}}_{\text{ext}}(\tau)d\tau\,, \tag{5}$$

where $\tilde{\boldsymbol{f}}_{\text{ext}} = \boldsymbol{f}_{\text{d}} - \boldsymbol{f}_{\text{ext}}$, and $\boldsymbol{K}_{\text{p}}, \boldsymbol{K}_{\text{i}} \in \mathbb{R}^{6\times6}$ are diagonalized proportional and integral gain matrices. The final desired joint torques $\boldsymbol{\tau}_{\text{robot}} \in \mathbb{R}^n$ are computed as

$$\boldsymbol{\tau}_{\text{robot}} = \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{f}_{\text{robot}}\,, \tag{6}$$

where $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{6\times n}$ is the robot Jacobian.

This unified control scheme allows precise and compliant execution of BOs. However, contact-rich processes demand more than just local feedback. We therefore integrate global process awareness using context-aware policy blending, described next.

C. Tactile Memory for Continuous Policy Blending

In contact-rich manipulation, process stages do not follow fixed durations. The robot must infer, from real-time interaction, when to transition between BOs. We address this with a tactile memory-driven policy blending strategy that monitors the evolution of a sensory state vector $\boldsymbol{s}_{\text{t}}$ and predicts BO termination in a data-driven yet physically grounded manner.

*1) **Process Quality Metrics**:* Each sensory state $\boldsymbol{s}_{\text{t}}$ comprises 32 Process Quality Metrics (PQMs), reflecting physical, kinematic, and energetic states. These include: end-effector pose $\boldsymbol{x} \in \mathbb{R}^6$, velocity $\dot{\boldsymbol{x}} \in \mathbb{R}^6$, external forces $\boldsymbol{f}_{\text{ext}} \in \mathbb{R}^6$, measured joint torques $\boldsymbol{\tau}_j \in \mathbb{R}^7$, the end-effector's distance $\tilde{\boldsymbol{x}}_{\text{dist}} \in \mathbb{R}^3$ to the goal $\boldsymbol{x}_{\text{goal}} \in \mathbb{R}^3$ in translational directions ($\boldsymbol{x} - \boldsymbol{x}_{\text{goal}} = \tilde{\boldsymbol{x}}_{\text{dist}}$) and its magnitude $\|\tilde{\boldsymbol{x}}_{\text{dist}}\|$, kinetic energy E_{KE} and potential energy E_{PE} of the tool atthat reflect the robot'she end-effector, as well as the time derivative $\dot{\text{E}}_{\text{PE}}$ of E_{PE}. The full sensory state vector $\boldsymbol{s}_{\text{t}} \in \mathbb{R}^{32}$ at each timestep $t \in \mathbb{R}$ is

$$\boldsymbol{s}_{\text{t}} = \left[\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{f}_{\text{ext}}, \boldsymbol{\tau}_j, \tilde{\boldsymbol{x}}_{\text{dist}}, \|\tilde{\boldsymbol{x}}_{\text{dist}}\|, \text{E}_{\text{KE}}, \text{E}_{\text{PE}}, \dot{\text{E}}_{\text{PE}}\right], \tag{7}$$

where the kinetic energy of the tool is given by

$$\text{E}_{\text{KE}} = \frac{1}{2}m_t\dot{\boldsymbol{x}}^T\dot{\boldsymbol{x}}\,, \tag{8}$$

being m_t is the mass of the end-effector tool. The potential energy of the tool is

$$\text{E}_{\text{PE}} = m_t\boldsymbol{g}^T\boldsymbol{x}_{\text{dist}}\,, \tag{9}$$

with $\boldsymbol{g} \in \mathbb{R}^3$ the gravity vector. The interplay between these energies reveals key transitions: a sharp drop in E_{KE} might indicate contact, while changes in E_{PE} often signify vertical motion, e.g., tool insertion or lift-off. To highlight the dynamics of such transitions, the time derivative $\dot{\text{E}}_{\text{PE}}$ is included, as it emphasizes moments of rapid energy change typically associated with making or breaking contact.

*2) **BOs Success Decision**:* A BiLSTM network encodes $\boldsymbol{s}_{\text{t}}$, capturing temporal dependencies both forward and backward in time [27]. We define the forward hidden state at time t as $\overrightarrow{\boldsymbol{h}}_t \in \mathbb{R}^d$ and the backward hidden state as $\overleftarrow{\boldsymbol{h}}_t \in \mathbb{R}^d$, which are updated as

$$\overrightarrow{\boldsymbol{h}}_t = \text{LSTM}_f(\boldsymbol{s}_{\text{t}}, \overrightarrow{\boldsymbol{h}}_{t-1}), \tag{10}$$

$$\overleftarrow{\boldsymbol{h}}_t = \text{LSTM}_b(\boldsymbol{s}_{\text{t}}, \overleftarrow{\boldsymbol{h}}_{t+1})\,, \tag{11}$$

where LSTM_f and LSTM_b denote the forward and backward LSTM functions, respectively. The concatenated hidden state $\boldsymbol{h}_t \in \mathbb{R}^{2d}$

$$\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}}_t; \overleftarrow{\boldsymbol{h}}_t]\,, \tag{12}$$

which serves as the latent memory of the process at time t. A fully connected readout layer, parameterized by weights $\mathbf{W} \in \mathbb{R}^{1\times2d}$ and bias $\text{b} \in \mathbb{R}$, estimates the termination probability

$$\hat{y}_t = \sigma(\mathbf{W}\boldsymbol{h}_t + \text{b}), \tag{13}$$

where $\hat{y}_t \in (0, 1)$ and $\sigma(\cdot)$ is the sigmoid activation. The model is trained using a binary cross-entropy loss function.

To reduce false positives caused by transient pauses, e.g., when the tool momentarily stalls during a BO, we smooth $\hat{y}_t$ using a moving average of length k and compare it against a threshold $\delta \in (0, 1)$. The BO is considered complete only if the smoothed $\hat{y}_t > \delta$.

*3) **Smooth Blending Between BOs**:* Each BO is associated with a scalar blending weight. These weights determine the contribution of each BO's control output to the final control command. At any moment, the robot either executes a single BO with its corresponding weight set to 1 and all others set to 0, or it smoothly blends between two BOs by interpolating their weights based on the robot's internal state (e.g., detecting contact or reaching a position). The blending phase starts at time t_{blend}, where $t_{\text{blend}} \in \mathbb{R}$ denotes the time when blending starts. During blending, the normalized blending time $t_{\text{norm}} \in [0, 1]$ is defined as

$$t_{\text{norm}} = \frac{t_c - t_{\text{blend}}}{T_{\text{blend}}}, \tag{14}$$

where $t_c \in \mathbb{R}$ is the current time, and $T_{\text{blend}} > 0$ is the total blending duration. A smooth sigmoid-like blending coefficient $\alpha \in [0, 1]$ is computed as

$$\alpha = \frac{1}{1 + \exp\left(-l\left(t_{\text{norm}} - 0.5\right)\right)}, \tag{15}$$

where $l > 0$ controls the *sharpness of the transition*. The blending weights for the current and next BO are then defined complementarily as

$$w_{\text{current}} = 1 - \alpha, \tag{16}$$

$$w_{\text{next}} = \alpha, \tag{17}$$

where $w_{\text{current}}, w_{\text{next}} \in [0, 1]$ are the *scalar blending weights* corresponding to the current and next BO, respectively. All other BO weights are set to zero during the blend. For instance, blending from *establish contact* to *find socket* implies

$$w_{\text{contact}} = 1 - \alpha\,, \tag{18}$$

$$w_{\text{socket}} = \alpha\,, \tag{19}$$

where w_{contact} and w_{socket} denote the blending weights of the *establish contact* and *find socket* BOs. During this blending phase, other BOs' weights are set to zero. Furthermore, the blended control command to be sent to the robot $\boldsymbol{f}'_{\text{ctrl}} \in \mathbb{R}^6$ is computed as the weighted sum of each BO's control command $\boldsymbol{f}_{\text{ctrl,BO}} \in \mathbb{R}^6$, weighted by their respective blending weights $w_{\text{BO}} \in [0, 1]$, where BO ranges over all BOs active in the process

$$\boldsymbol{f}'_{\text{ctrl}} = \sum_{\text{BOs}} w_{\text{BO}} \boldsymbol{f}_{\text{ctrl,BO}}. \tag{20}$$

Explicitly, for the USB plugging and unplugging example with BOs *move-to-pose*, *establish contact*, *find socket*, *plug*, and *unplug*, the corresponding shortened forms (*go*, *contact*, *socket*, *plug*, *unplug*) are used, and the control input is given by

$$\begin{aligned}\boldsymbol{f}'_{\text{ctrl}} = {} & w_{\text{go}} \boldsymbol{f}_{\text{ctrl,go}} + w_{\text{contact}} \boldsymbol{f}_{\text{ctrl,contact}} \\ & + w_{\text{socket}} \boldsymbol{f}_{\text{ctrl,socket}} + w_{\text{plug}} \boldsymbol{f}_{\text{ctrl,plug}} \\ & + w_{\text{unplug}} \boldsymbol{f}_{\text{ctrl,unplug}},\end{aligned} \tag{21}$$

where each w_{BO} is the blending weight and $\boldsymbol{f}_{\text{ctrl,BO}}$ is the control output vector corresponding to the basic operation BO.

This mechanism enables state-triggered and smooth transitions between BOs. Blending outputs smoothly instead of abruptly switching supports continuous and functionally safe control behavior.

D. Energy Tanks for Functional Safety

While unified force-impedance control provides compliant tracking, it does not inherently ensure passivity [28]. If the controller injects excessive energy, stability may be lost during unforeseen contact events while blending the control commands. We enforce functional safety by regulating energy injection through a virtual energy tank to mitigate this.

*1) **Passivity Condition and Risk of Instability**:* The robot's total kinetic energy is

$$\mathrm{E}_{\text{robot}} = \frac{1}{2} \dot{\boldsymbol{x}}^T \boldsymbol{M}_{\text{C}} \dot{\boldsymbol{x}}. \tag{22}$$

Taking the derivative and applying robot dynamics

$$\dot{\mathrm{E}}_{\text{robot}} = \dot{\boldsymbol{x}}^T \left(\boldsymbol{f}_{\text{robot}} + \boldsymbol{f}_{\text{ext}} - \boldsymbol{C}_{\text{C}} \dot{\boldsymbol{x}} - \boldsymbol{f}_{\text{g}} + \tfrac{1}{2} \dot{\boldsymbol{M}}_{\text{C}} \dot{\boldsymbol{x}} \right). \tag{23}$$

Due to the skew-symmetry of $(\dot{\boldsymbol{M}}_{\text{C}} - 2\boldsymbol{C}_{\text{C}})$, it holds that $\dot{\boldsymbol{x}}^T (\dot{\boldsymbol{M}}_{\text{C}} - 2\boldsymbol{C}_{\text{C}}) \dot{\boldsymbol{x}} = 0$. Substituting $\boldsymbol{f}_{\text{robot}} = \boldsymbol{f}'_{\text{ctrl}} + \boldsymbol{f}_{\text{g}}$ yields

$$\dot{\mathrm{E}}_{\text{robot}} = \dot{\boldsymbol{x}}^T \boldsymbol{f}'_{\text{ctrl}} + \dot{\boldsymbol{x}}^T \boldsymbol{f}_{\text{ext}}\,. \tag{24}$$

When the environment is passive ($\mathrm{P}_{\text{ext}} = \dot{\boldsymbol{x}}^T \boldsymbol{f}_{\text{ext}} \leq 0$), the term $\dot{\boldsymbol{x}}^T \boldsymbol{f}'_{\text{ctrl}}$ must also be negative to preserve passivity. Since this is not guaranteed, we introduce an energy tank to cap the robot's injected power into the environment.

*2) **Virtual Energy Tank Mechanism**:* We define a virtual energy tank E_{tank} initialized with $\mathrm{E}_{\text{tank}}(0) > 0$. Its evolution is governed by the power dynamics $\dot{\mathrm{E}}_{\text{tank}}$

$$\dot{\mathrm{E}}_{\text{tank}} = -\mathrm{P} = -\dot{\boldsymbol{x}}^T \boldsymbol{f}'_{\text{ctrl}}\,, \tag{25}$$

i.e., the tank depletes when the robot injects power P into the environment. We introduce a valve function $\rho(\mathrm{E}_{\text{tank}})$

$$\rho(\mathrm{E}_{\text{tank}}) = \begin{cases} 1, & \mathrm{E}_{\text{tank}} > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{26}$$

that blocks control effort once the tank is empty. The robot's control wrench is then

$$\boldsymbol{f}_{\text{robot}} = \rho \boldsymbol{f}'_{\text{ctrl}} + \boldsymbol{f}_{\text{g}}. \tag{27}$$

As a result, the total energy rate satisfies

$$\dot{\mathrm{E}}_{\text{robot}} + \dot{\mathrm{E}}_{\text{tank}} = (\rho - 1)\mathrm{P} + \mathrm{P}_{\text{ext}} \leq 0, \tag{28}$$

ensuring that the controller cannot inject unbounded energy.

The final closed-loop dynamics of the robot become

$$\boldsymbol{M}_{\text{C}}(\boldsymbol{q}) \ddot{\boldsymbol{x}} + \boldsymbol{C}_{\text{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{x}} - \rho \boldsymbol{f}'_{\text{ctrl}} - \boldsymbol{f}_{\text{ext}} = 0\,. \tag{29}$$

Next, we design the experiments to present our pipeline's efficacy over the baseline methods.

III. Experimental Validation

Experiments use a 7-DOF Franka Emika Research robot, controlled at 1 kHz, running Ubuntu 20.04 with a real-time kernel, as in Fig. 3. The robot's internal proprioceptive sensors measure forces and motion.

TABLE I: **Library of Basic Operations (BOs)**. The notation $\mathrm{diag}[a, b, c, \dots]$ represents a diagonal matrix with elements $a, b, c, \dots$ along its diagonal. c.d. is short for critical damping.

BOs	Policy Description	Control Parameters
BO1—Move-to-pose	Smoothly reach a target pose $\boldsymbol{x}_d$	$\boldsymbol{K}_{\mathrm{C}} = \mathrm{diag}[700, 700, 700, 50, 50, 50]$, $\boldsymbol{D}_{\mathrm{C}}$: c.d., $\boldsymbol{K}_{\mathrm{p}} = \boldsymbol{K}_{\mathrm{i}} = 0.5\boldsymbol{I}_6$
BO2—Establish contact	Advance until contact is detected $\boldsymbol{x}_d$	$\boldsymbol{K}_{\mathrm{C}} = \mathrm{diag}[700, 700, 700, 50, 50, 50]$, $\boldsymbol{D}_{\mathrm{C}}$: c.d., $\boldsymbol{K}_{\mathrm{p}} = \boldsymbol{K}_{\mathrm{i}} = 0.5\boldsymbol{I}_6$
BO3—Find socket	Spiral search under contact to locate the socket $\boldsymbol{x}_d, \boldsymbol{f}_d$	$\boldsymbol{K}_{\mathrm{C}} = \mathrm{diag}[450, 450, 0, 10, 10, 8]$, $\boldsymbol{D}_{\mathrm{C}}$: c.d., $\boldsymbol{K}_{\mathrm{p}} = \boldsymbol{K}_{\mathrm{i}} = 0.5\boldsymbol{I}_6$
BO4—Find hole	Spiral search under contact to locate the hole $\boldsymbol{x}_d, \boldsymbol{f}_d$	$\boldsymbol{K}_{\mathrm{C}} = \mathrm{diag}[700, 700, 0, 50, 50, 50]$, $\boldsymbol{D}_{\mathrm{C}}$: c.d., $\boldsymbol{K}_{\mathrm{p}} = \boldsymbol{K}_{\mathrm{i}} = 0.5\boldsymbol{I}_6$
BO5—Find screw	Align to screw head by applying force $\boldsymbol{f}_d$	$\boldsymbol{D}_{\mathrm{C}}$: c.d., $\boldsymbol{K}_{\mathrm{p}} = \boldsymbol{K}_{\mathrm{i}} = 0.5\boldsymbol{I}_6$
BO6—Insert	Peg insertion with wiggling $\boldsymbol{f}_d$	$\boldsymbol{D}_{\mathrm{C}}$: c.d., $\boldsymbol{K}_{\mathrm{p}} = \boldsymbol{K}_{\mathrm{i}} = 0.5\boldsymbol{I}_6$
BO7—Screw	Push-to-trigger screwdriver using constant downward force $\boldsymbol{f}_d$	$\boldsymbol{D}_{\mathrm{C}}$: c.d., $\boldsymbol{K}_{\mathrm{p}} = \boldsymbol{K}_{\mathrm{i}} = 0.5\boldsymbol{I}_6$
BO8—Plug	Force-driven plug with wiggling $\boldsymbol{f}_d$	$\boldsymbol{D}_{\mathrm{C}}$: c.d., $\boldsymbol{K}_{\mathrm{p}} = \boldsymbol{K}_{\mathrm{i}} = 0.5\boldsymbol{I}_6$
BO9—Unplug	Intermittent pulling force $\boldsymbol{f}_d$	$\boldsymbol{D}_{\mathrm{C}}$: c.d., $\boldsymbol{K}_{\mathrm{p}} = \boldsymbol{K}_{\mathrm{i}} = 0.5\boldsymbol{I}_6$

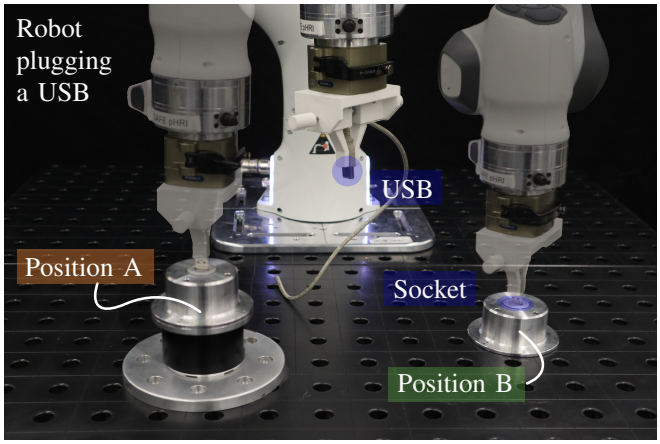

Fig. 3: **Experimental setup and protocol.** We compare threshold-based switching to our tactile memory-driven policy blending across USB plug/unplug, two peg tolerances, and two screw sizes, each executed at two positions, i.e, A and B, with 10 trials per position. We report repeatability in success compared to the baseline and transferability.

A. Data Collection, Labeling, and Training

We recorded successful executions for a subset of BOs listed in Table I, to be used across processes, i.e., peg insertion, USB plugging and unplugging, and screw-driving, for different starting positions and hole, socket, and screw locations, keeping the per-BO count below 40 to capture representative PQMs variability, e.g., force spikes, and potential and kinetic energy evolutions. Each run logs 32 PQMs at 1 kHz, which are segmented into fixed-length windows of 500, which form the BiLSTM input.

When labeling the data, we look for sudden changes in force, potential energy, or kinetic energy to determine when BOs end. For instance, in the BO of establishing contact, we observe a sudden force spike along the z direction with a stop in changes in $\mathrm{E_{KE}}$ and $\mathrm{E_{PE}}$. Similarly, at the end of the screwing process, the screw stops rotating, and the torque from the electronic screwdriver is transferred to the end-effector. At this point, we see a spike in the rotational forces, and the kinetic energy drops because the screw is no longer being driven.

The model's architecture consists of an input layer, two BiLSTM layers with 64 hidden units, and an output layer. Each model is trained for up to 100 epochs with a learning rate of 0.001, using binary cross-entropy as the loss function. To reduce sensitivity to noise and variations in BOs, we apply a fixed-size moving average to the recent outputs and use BO-centric thresholds, as listed in Table II. These parameters are set once and remain fixed thereafter.

TABLE II: BiLSTM operational parameters.

Model	Threshold	Moving Average Size
Establish Contact	0.90	2
Find Hole	0.90	2
Find Socket	0.85	3
Plug	0.80	8
Screw-driving	0.90	4
Insert	0.80	8

B. Experimental Procedure

We test robustness and transferability under variations in i.) spatial location, ii.) geometry/tolerance (pegs), and iii.) size (screws). For each of five process variants (USB, small screw, big screw, high-tolerance peg, low-tolerance peg), we run ten trials at Position A $\boldsymbol{x}_A$ and ten at Position B $\boldsymbol{x}_B$, for a total of 100 trials

$$\boldsymbol{x}_A = [0.465, -0.156, 0.094, 0.962, -0.275, -0.002]^T,$$
$$\boldsymbol{x}_B = [0.381,\ \ 0.192, 0.025, 0.955, -0.297, -0.003]^T,$$

which are chosen to differ from the training positions.

The desired force/motion policies and control parameters are fixed during the experiments; no retraining or tuning is performed. Each process executes as a BO sequence under unified force-impedance control; BiLSTMs detect BO termination, and a sigmoid smoothly blends the control wrenches between BOs. The shared BOs of the move-to-pose and establish contact are reused across all processes.

*1) **Experiment I—Peg insertion**:* A peg is inserted into a hole using the sequence of the following BOs: move-to-pose, establish contact, find hole, and insert. Two clearance-fit pegs: low-tolerance (tight, $<0.03\,\mathrm{mm}$; height $29.91\,\mathrm{mm}$; diameter of $6\,\mathrm{mm}$) and high-tolerance (loose, $<0.07\,\mathrm{mm}$; height $30.85\,\mathrm{mm}$; diameter of $3\,\mathrm{mm}$).

*2) **Experiment II—USB plug and unplug**:* USB-A plug is plugged into a socket and unplugged from it by the following BOs: move-to-pose, establish contact, find socket, plug, and unplug.

*3) **Experiment III—Screw-driving**:* Screw-driving is done by the following BOs: move-to-pose, establish contact, find the screw head, and screw-drive. We use a push-to-trigger screwdriver, which requires 25 N, and perform the process for $\mathrm{M4{\times}12\,mm}$, namely small, and $\mathrm{M8{\times}16\,mm}$, big, screws.

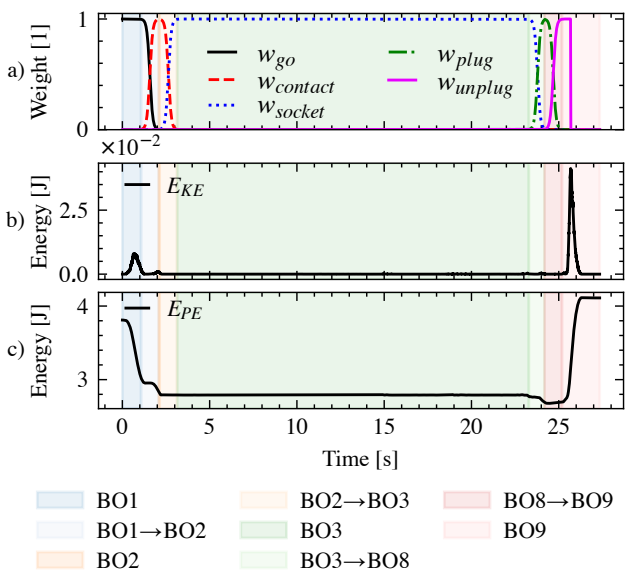

Fig. 4: **USB plug-unplug.** a) BO blending weights formed after BiLSTM-detected success (soft transitions via sigmoid). b) Kinetic energy $\mathrm{E_{KE}}$ and c) potential energy $\mathrm{E_{PE}}$ of the tool; contact and phase changes align with energy evolutions. During *find socket*, since the tool maintains contact with the surface and moves slowly to avoid missing the socket, both $\mathrm{E_{PE}}$ and $\mathrm{E_{KE}}$ remain nearly constant, with $\mathrm{E_{KE}}$ staying low due to the minimal motion.

All the experiments for the variants are conducted ten times, first at Position A and then at Position B.

*4) **Baseline**:* Our baseline uses the same basic operations, but instead of blending them during transitions, it immediately moves to the next operation. Furthermore, instead of using trained models to detect when an operation is finished, position-based thresholds are used to decide when to switch.

We use position-based thresholds rather than time-based ones because the completion of a BO for a contact-rich process is not deterministic in duration. For instance, in finding the hole BO, the time to completion depends on where the end-effector first contacts the surface, which varies across executions. Therefore, using time-based thresholds would not reliably indicate process completion.

C. Performance Metrics

*1) **Repeatability in success**:* For the baseline, we tune position thresholds once until a successful run, then measure success over 10 repetitions and compare to our pipeline for only at Position A.

*2) **Transferability**:* We assess whether the same blending model, trained on nominal BO executions, successfully transfers to spatially shifted or physically varied process settings, i.e., peg tolerance, screw size. A run is deemed successful if all BO transitions are predicted correctly and the process is completed. We use move-to-pose and establish contact BOs across all processes to demonstrate that the same models can be successfully applied to different process variants and spatial settings.

A failure is recorded when the model prematurely transitions to the next phase, stalls due to incorrect PQM interpretation, or fails to terminate a BO. Importantly, we do not attribute failure to hardware or external disturbances unless the blending model causes incorrect termination.

In the next section, we present and analyze the results.

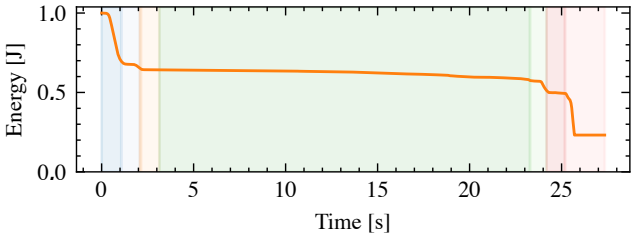

Fig. 5: **System's stability via energy tank evolution.** The tank is initialized at $\mathrm{E_{tank}}(0) = 1\,\mathrm{J}$ and never depletes below zero, enforcing passivity during contact-rich blending.

IV. Results and Discussion

A successful example from the USB plugging-and-unplugging experiments, showing when the model decides to terminate the BOs, is depicted in Fig. 4. The process begins with *move-to-pose*; as the end-effector accelerates toward the target, E_{KE} rises while E_{PE} decreases due to the lower socket height, as shown in Fig. 4b and Fig. 4c. Upon *establish contact*, E_{KE} momentarily drops, marking contact onset. Subsequent phases (*find socket*, *plug*, *unplug*) exhibit gradual BO weight changes with no hard switches, and the composed wrench remains continuous. During *find socket*, the tool maintains surface contact and moves slowly, keeping both E_{PE} and E_{KE} nearly constant, with E_{KE} remaining low due to the slow motion.

The controller's stability is maintained via a virtual energy tank. It is initialized such that $\mathrm{E_{tank}}(0) = 1\,\mathrm{J}$, and never drops below zero during the process, as illustrated in Fig. 5.

A. Repeatability in Success

Even with the given thresholds, the baseline method's success rate still falls below that of our proposed method, as shown in Fig. 6. In the USB plugging process, for instance, after fine-tuning the baseline, 4 out of 10 executions succeeded. In contrast, in our proposed method, without any further refinements, the success rate is eight out of ten.

In the case of screw-driving, we define success as the full screw-driving of the screw into the screw hole, which is the height of the screw. Using this goal definition, the success rate is $0\,\%$ for the baseline scenario. However, $50\,\%$ of the cases are within the goal position by the error of approximately $2\,\mathrm{mm}$. Thus, we update the success definition to be within the goal position by the baseline method's threshold of $2\,\mathrm{mm}$. We observe that fine-tuning the goal positions can be challenging for the baseline with full screw-driving. In other words, if the threshold is set high, the screw fails to fully insert; if it is set low, the robot keeps applying force, which is dangerous because the screw could be locked in the hole, resulting in large torques at the robot's end effector. Nevertheless, with our method, without any further training or fine-tuning, the robot achieves full screw driving with a $100\,\%$ success rate.

Furthermore, in the absence of our method and with hard switching using fine-tuned parameters (i.e., position data), the USB plugging process required 10 fine-tuning steps to

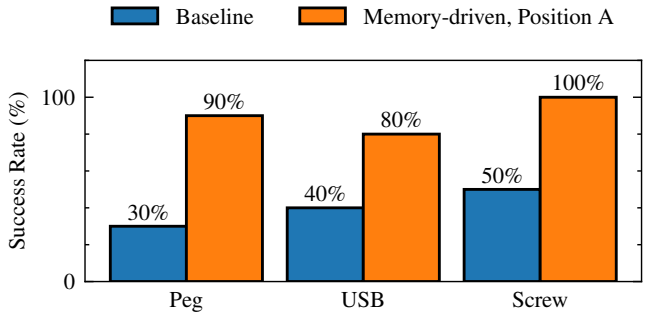

Fig. 6: **Success rate comparison.** Tactile memory-driven continuous policy blending in unified force-impedance control improves repeatability in success over position threshold-based switching for all evaluated processes.

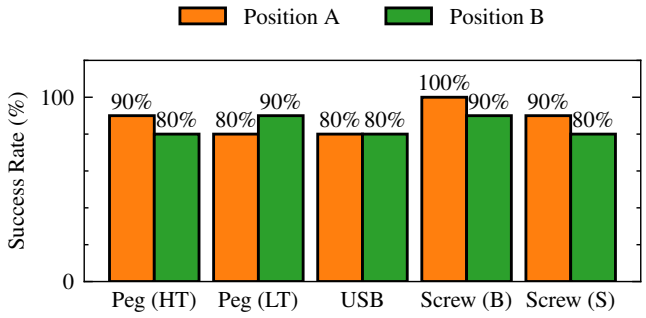

Fig. 7: **Transfer across location and object variants.** Success rates at positions A and B for USB, two peg tolerances, such as high tolerance (HT) and low tolerance (LT), and two screw sizes, namely big (B) and small (S), using the *same* BO-specific models and controller parameters.

TABLE III: Completion Times (CT) for Different Processes

Process	**Mean CT [s]**	**Std CT [s]**
USB, Tactile memory-driven	32.683	10.596
USB, Baseline	25.907	13.278
Screw, Tactile memory-driven	4.726	1.103
Screw, Baseline	3.084	0.508
Peg, Tactile memory-driven	32.227	7.088
Peg, Baseline	52.614	10.817

complete successfully. Even when the robot starts working with the given parameters, the process needs to be repeated due to position disturbances, resulting in unnecessary time loss to complete it successfully.

Additionally, as shown in Table III, the USB and screwing processes in the baseline are completed in less time than in our method. However, even though they are faster, the success rates drop significantly compared to the tactile memory-driven continuous policy blending. Also, in the peg-in-hole process, the baseline's completion time is longer than that of the tactile memory-driven approach.

We can deduce that screw-driving is more reliable in the tactile memory-driven method with higher success rates, i.e., 80-100 %, while USB plugging and peg-in-hole remain slightly more challenging, but success rates are still high, namely 80-90 %.

Additionally, failures were observed in situations where PQMs may resemble the end of a BO. In detail, if the USB becomes stuck in the socket during plugging, the resulting forces and lack of movement are similar to those in the plugged-in state, leading to misclassification as completed. One option to overcome this would be to use an external perception, e.g., a camera, which could significantly enhance the performance across all processes by eliminating these errors. It also helps monitor BO completions externally.

B. Transferability

Performance is consistently high, i.e., >80 %, across four processes with five variants for both positions, as shown in Fig. 7. Position A and B differences are relatively minor, even though Position B shows slightly lower success rates than Position A in three processes out of five. The models, which are trained for a maximum of 40 runs per BO and collected at locations different from the experiments, achieve high success rates across different contact tolerances, spatial locations, and tool sizes. This provides a straightforward platform for executing the processes. When a human operator wants the robot to execute a process, they only need to select which process to execute and the position where it will be executed. Then the robot can directly start executing the selected process as a sequence of BOs in the library. Without the need for further refinements or fine-tuning, the trained models can decide when to terminate the current BOs using the history of PQMs and inferring success from the robot's current sensory data. Furthermore, it is noteworthy that the BOs of move-to-pose and establish contact are reused across processes without retraining. Thus, those BOs could transfer across positions within the same processes and across processes for different sizes and tolerances.

Overall, in terms of transferability, the position difference did not affect the outcomes of the processes; the success rates across positions remained close.

C. Final Remarks

Despite these promising results, the current framework assumes the library of basic operations and their ordering within a process are known in advance. This limits transferability to entirely new process types or unstructured environments. Also, if one BO completes prematurely, the method does not backtrack for the current implementation, which could be an interesting research direction to investigate further.

Additionally, we were unable to teach the robot to detect when the end effector is aligned with the screw head using onboard sensors because the force magnitudes are too close to the sensor noise level, and the force signal shows almost no movement. As a result, very minimal events are challenging to teach without external perception, such as a camera.

V. Conclusion

Industrial robot programming often relies on rigid, predefined transitions for basic operations, hindering transferability in contact-rich, multimodal processes such as insertion, plugging, and screw-driving. To overcome these limitations, we introduced a tactile memory for continuous policy blending in unified force-impedance control, enabling robots to adapt execution strategies based on prior experience and real-time sensor feedback.

Our approach decomposes each manipulation process into reusable basic operations (BOs), each of which is annotated with 32 process quality metrics (PQMs) that describe the robot's physical interaction state. A bidirectional LSTM processes temporal PQM sequences to predict operation success; afterward, soft blending weights are generated, enabling continuous activation of BOs under a unified force-impedance controller. Functional safety is enforced through a virtual energy tank mechanism that constrains power exchange and ensures system passivity. Our experimental results across four industrial manipulation processes validate the effectiveness of this framework, demonstrating smoother transitions, improved robustness, and generalization across spatial and tool variations.

Future work will explore merging tactile perception with visual feedback to enrich the PQMs and enable multimodal grounding of manipulation behavior.

This work contributes toward bridging the gap between rigid industrial automation and human-like intelligent manipulation, paving the way for flexible, context-aware robotic systems that operate robustly and adaptively without relying on human-defined step-by-step instructions.

VI. Acknowledgement

The authors acknowledge financial support from the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi) for the Lighthouse Initiative KI.FABRIK, (Phase 1: Infrastructure) and the support of the project SafeRoBAY (grant number: DIK0203/01). This work has received funding from the European Union's Horizon Europe research and innovation programme as part of the project FlexCycle under grant agreement No. 101189600.

References

[1] L. Ren, J. Dong, S. Liu, L. Zhang, and L. Wang, "Embodied intelligence toward future smart manufacturing in the era of ai foundation model," *IEEE/ASME Transactions on Mechatronics*, pp. 1–11, 2024.

[2] A. Lambert and S. Gupta, "Disassembly modeling for assembly, maintenance, reuse and recycling," 12 2004.

[3] L. Johannsmeier, M. Gerchow, and S. Haddadin, "A framework for robot manipulation: Skill formalism, meta learning and adaptive control," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5844–5850.

[4] L. Johannsmeier, S. Schneider, Y. Li, E. Burdet, and S. Haddadin, "A process-centric manipulation taxonomy for the organization, classification and synthesis of tactile robot skills," *Nature Machine Intelligence*, vol. 7, no. 6, pp. 916–927, June 2025. [Online]. Available: https://doi.org/10.1038/s42256-025-01045-3

[5] R. J. Kirschner, K. Karacan, A. Melone, and S. Haddadin, "Categorizing robots by performance fitness into the tree of robots," *Nature Machine Intelligence*, vol. 7, no. 3, pp. 459–470, Mar. 2025. [Online]. Available: https://doi.org/10.1038/s42256-025-00995-y

[6] C. Zieliński and T. Winiarski, "Motion Generation in the MRROC++ Robot Programming Framework," *The International Journal of Robotics Research*, vol. 29, no. 4, pp. 386–413, 2010, _eprint: https://doi.org/10.1177/0278364909348761. [Online]. Available: https://doi.org/10.1177/0278364909348761

[7] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, and P. Fraisse, "Collaborative manufacturing with physical human–robot interaction," *Robotics and Computer-Integrated Manufacturing*, vol. 40, pp. 1–13, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584515301769

[8] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 620–629, March 2016.

[9] F. Kulakov, G. V. Alferov, P. Efimova, S. Chernakova, and D. Shymanchuk, "Modeling and control of robot manipulators with the constraints at the moving objects," in *2015 International Conference "Stability and Control Processes" in Memory of V.I. Zubov (SCP)*, Oct 2015, pp. 102–105.

[10] F. Ficuciello, L. Villani, and B. Siciliano, "Variable impedance control of redundant manipulators for intuitive human–robot physical interaction," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 850–863, Aug 2015.

[11] C. Ott, A. Dietrich, and A. Albu-Schäffer, "Prioritized multi-task compliance control of redundant manipulators," *Automatica*, vol. 53, pp. 416–423, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109815000163

[12] J. D. Schutter, T. D. Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007.

[13] P. Pastor, M. Kalakrishnan, L. Righetti, and S. Schaal, "Towards associative skill memories," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, Nov 2012, pp. 309–315.

[14] T. Migimatsu and J. Bohg, "Object-centric task and motion planning in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 844–851, 2020.

[15] M. Kim, S. Niekum, and A. D. Deshpande, "Scape: Learning stiffness control from augmented position control experiences," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 1512–1521. [Online]. Available: https://proceedings.mlr.press/v164/kim22b.html

[16] J. Liang, X. Cheng, and O. Kroemer, "Learning preconditions of hybrid force-velocity controllers for contact-rich manipulation," 2022.

[17] K. Burns, A. Jain, K. Go, F. Xia, M. Stark, S. Schaal, and K. Hausman, "Genchip: Generating robot policy code for high-precision and contact-rich manipulation tasks," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 9596–9603.

[18] H. Zhang, G. Solak, G. J. G. Lahr, and A. Ajoudani, "Srl-vic: A variable stiffness-based safe reinforcement learning for contact-rich robotic tasks," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5631–5638, 2024.

[19] X. Chen, T. Ni, K. Karacan, H. Sadeghian, and S. Haddadin, "Online transfer and adaptation of tactile skill: A teleoperation framework," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=6X3ybeVpDi

[20] T. Narita and O. Kroemer, "Policy blending and recombination for multimodal contact-rich tasks," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2721–2728, 2021.

[21] M. Noseworthy, B. Tang, B. Wen, A. Handa, C. Kessens, N. Roy, D. Fox, F. Ramos, Y. Narang, and I. Akinola, "Forge: Force-guided exploration for robust contact-rich manipulation under uncertainty," *IEEE Robotics and Automation Letters*, vol. 10, no. 5, pp. 4436–4443, 2025.

[22] L. Halt, F. Pan, P. Tenbrock, A. Pott, and T. Seel, "A transferable force controller based on prescribed performance for contact establishment in robotic assembly tasks," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 830–835.

[23] M. S. Finkbeiner, T. R. Schaefle, J. Beck, J. T. Stoll, W. Kraus, and M. Giftthaler, "A concept for unifying the performance assessment of industrial robot systems with closed-loop dynamic trajectories," in *ISR Europe 2023; 56th International Symposium on Robotics*, 2023, pp. 398–404.

[24] M. Hwang, J. Hejna, D. Sadigh, and Y. Bisk, "Motif: Motion instruction fine-tuning," *IEEE Robotics and Automation Letters*, vol. 10, no. 3, pp. 2287–2294, 2025.

[25] N. Hogan, "Impedance control: An approach to manipulation: Part ii—implementation," 1985.

[26] M. H. Raibert and J. J. Craig, "Hybrid position-force control of manipulators," *book*, 1981.

[27] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional lstm networks for improved phoneme classification and recognition." 01 2005, pp. 799–804.

[28] S. Haddadin and E. Shahriari, "Unified force-impedance control," *The International Journal of Robotics Research*, vol. 43, no. 13, pp. 2112–2141, 2024. [Online]. Available: https://doi.org/10.1177/02783649241249194