SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY

TECHNICAL UNIVERSITY OF MUNICH

Nudging Human Decision-Making in Teleoperation Robotics - Designing an Artificial Recommender System for Optimal and Controlled Operator Guidance

Thesis for the attainment of the academic degree
Master of Science (M.Sc.)

Author:	Alexander Wegener
Supervisor:	Prof. Dr. Alexander König
	Chair of Robotics and Systems Intelligence
Advisor:	Lingyun Chen, M. Sc.
Submission Date:	December 13, 2025, Garching bei München

I confirm that this master's thesis is my own work and I have documented all sources and material used.

München, December 13, 2025 Alexander Wegener

Abstract

Teleoperated manipulation enables robots to perform contact-rich tasks in unstructured or hazardous environments, but operator performance is often limited by workload, imperfect perception, and the need to make rapid decisions under uncertainty. This thesis investigates whether *nudging*—subtle, choice-preserving guidance delivered through the interface—can improve teleoperation performance without undermining the operator's sense of control. We present a VR-based teleoperation system that combines a first-person, head-coupled visual interface with a 6-DoF haptic device rendering contact and assistance forces. On top of this interface, we implement an adaptive recommender based on contextual bandits that selects between informational nudges (e.g., task-relevant cues and recommendations) and dynamic nudges that modulate continuous interaction parameters, such as workspace scaling, guidance strength, and resistive haptic damping, to shape motion and contact behavior.

The approach is evaluated in a user study on two representative manipulation tasks: a toy gearbox assembly and a contact-rich peg-in-hole insertion. Across assistance modes ranging from manual operation to combined informational and dynamic nudging, we quantify efficiency, interaction quality, and perceived workload using task time, success metrics, motion/force statistics, and NASA-TLX. The results indicate that added guidance and information reduce abrupt corrections and high interaction peaks while maintaining comparable nominal operating speeds, and that perceived mental demand and frustration decrease as assistance is introduced. The thesis discusses the trade-offs between improved smoothness and preserving operator agency, and outlines limitations and future directions for learning personalized, context-aware assistance in teleoperated manipulation.

Contents

Acknowledgements

1 Introduction

Teleoperation is a critical technology in robotics, enabling human operators to control remote systems in complex, unstructured, or hazardous environments. By combining human cognitive skills with the robot's physical capabilities, teleoperation is essential for tasks where full autonomy is currently infeasible. However, teleoperation systems face fundamental challenges, including communication latency, limited sensory feedback, and the high cognitive load placed on the operator. These factors often lead to operator fatigue, reduced situational awareness, and ultimately, a degradation of performance, reliability, and efficiency.

To address these limitations, **shared control** paradigms are employed to distribute control authority between the human and an autonomous system. This thesis investigates a subtle yet powerful strategy within shared control: **informational nudging**. Derived from decision theory, nudging is the principle of steering human behavior toward safer or more optimal outcomes without removing the operator's fundamental freedom of choice.

The core contribution of this work is the design and validation of an **Artificial Recommender System** that implements this nudging philosophy to provide optimal and controlled guidance to the human operator during teleoperation tasks. The system is built upon a theoretical foundation that models the operator's behavior under the constraints of **bounded rationality**, acknowledging that humans use heuristics and limited information rather than fully maximizing expected utility, especially under pressure.

1.1 Research and System Design

The proposed system integrates a high-transparency teleoperation architecture, comprising a **Franka Emika Panda** robotic arm, a **sigma.7** haptic interface for tactile feedback, and a **VR-based user environment**, connected via a dedicated, low-latency network.

The **Nudging Recommender System** utilizes a data-driven, adaptive framework based on the **contextual bandit** algorithm, which offers robust, computationally feasible, and consistently helpful recommendations. This approach overcomes the practical limitations identified in an initial, highly-modeled Stackelberg-inspired formulation. The recommender infers operator intent and translates its optimal policy into context-aware, personalized suggestions delivered via **dynamic nudges** (modifying haptic feedback parameters) and **informational nudges** (providing predictive or contextual cues).

1.2 Experimental Validation and Findings

The effectiveness of this guided-assistance system is evaluated across two distinct manipulation tasks: a sequential **Gearbox assembly** and a contact-rich and precision-critical **Tactile insertion task**. By comparing performance across various feedback modes from **Manual** (visual-only) to **Full** (comprehensive informational and dynamic nudging). The research rigorously quantifies the system's impact on operator behavior and performance.

Result analysis demonstrates that active assistance has a strong positive effect, particularly on the precision-critical insertion task. Active feedback modes (**Pure Info** and **Full**) are shown to substantially reduce contact forces, motion variability, and overall mental workload (**TLX** scores), indicating a more controlled and stable operation compared to the high-load, low-control characteristics of the Manual mode. Furthermore, **information-based modes** lead to the shortest task completion times for the sequential gearbox task, validating the system's ability to drive operators toward a more efficient execution strategy.

1.3 Thesis Outline

The remainder of this thesis is structured as follows:

- **Chapter 2: Background & State of the Art** reviews teleoperation, shared control, informational nudging, recommender systems, and models of human decision-making.
- **Chapter 3: System Architecture** details the distributed hardware and software framework, including the operator interface and the remote robotic avatar system.
- **Chapter 4: Nudging Recommender System** presents the formal modeling and implementation of the contextual bandit-based assistance system.
- **Chapter 5: Experimental Methodology** describes the tasks, feedback modes under comparison, participant procedure, and evaluation metrics.
- **Chapter 6: Results** presents the quantitative data and analysis of the experimental findings.
- **Chapter 7: Discussion** interprets the results and contextualizes the findings within the broader field of shared autonomy and human-robot interaction.
- **Chapter 8: Conclusion and Outlook** summarizes the main achievements and suggests directions for future research.

2 Background & State of the Art

2.1 Teleoperation and Shared Control

2.1.1 Foundations of Teleoperation

Teleoperation enables a human operator to remotely control a robot via a communication channel, often using a master–slave architecture where operator commands are translated into actuation signals for the robot, and sensory (e.g., visual, haptic) feedback is returned to the operator. The operator's cognitive skills and situational awareness are combined with the robot's physical capabilities — a key benefit for operating complex platforms in remote or hazardous environments. This paradigm applies especially well to humanoid robotics, where full autonomy remains challenging, and teleoperation retains a central role [12].

2.1.2 Challenges in Human–Robot Teleoperation

Despite progress, significant issues remain. Communication delay, limited bandwidth, and latency can degrade feedback transparency, leading to instability or oscillatory operator corrections. When feedback is restricted (e.g., visual-only), situational awareness becomes poor — particularly in cluttered or dynamic environments. Operators must maintain high cognitive load: monitoring feedback, planning motion, executing fine manipulations, and anticipating environmental changes [33]. As a result, performance depends heavily on operator skill, experience, and mental state; efficiency and reliability often degrade under stress, fatigue, or uncertainty.

2.1.3 Shared Control Paradigms

Shared control — or shared autonomy — seeks to alleviate the burden on operators by distributing control between human input and autonomous assistance. In such frameworks, the system arbitrates between human commands and robot autonomy, either blending inputs continuously or handing over control for subtasks when the autonomous module has higher confidence. Shared control reduces operator workload, improves stability, and makes complex tasks more tractable for non-expert operators. Recent literature provides a systematic classification of shared control strategies — including Semi-Autonomous Control (SAC), State-Guidance Shared Control (SGSC), and State-Fusion Shared Control (SFSC) — summarizing their respective trade-offs and open issues [30]. Also, implementations on redundant / whole-body robots have demonstrated viable shared control for complex, multi-DoF platforms [10].

2.1.4 Intelligent Assistance and Intent Prediction

Modern shared-autonomy systems increasingly include intent recognition and predictive assistance: by inferring the operator's likely goal or intended trajectory, the robot can proactively offer support — for instance, by planning safe motion paths, rejecting hazardous commands, or suggesting feasible alternatives. Probabilistic approaches (e.g., Bayesian filtering) have been shown effective to infer user goals from non-verbal cues and contextual data in assistive teleoperation settings [22, 42]. More recently, supervised-learning methods developed for mobile-robot teleoperation have demonstrated reliable online estimation of operator navigational goals in remote inspection and exploration tasks [53]. Building on such intent inference, robotic assistance becomes more anticipatory, adaptive, and personalized — enabling improved performance especially when sensory feedback or operator experience is limited.

2.2 Informational Nudging and Recommender Systems

2.2.1 Nudging Theory in Human–Machine Interaction

"Nudging" refers to influencing human decisions by shaping how choices are presented — without removing alternatives [29]. In human–machine systems, informational nudges can steer operator behavior by altering framing, highlighting certain options, reordering choices, or emphasizing recommended actions. This approach preserves operator autonomy but guides decisions toward safer or more efficient outcomes — a potentially powerful tool in high-stakes or cognitively demanding settings such as teleoperation.

2.2.2 Nudging in Robotics and Teleoperation

Though less explored than in consumer domains, recent work demonstrates robots employing nudge-like strategies to influence human behavior. For example, a study on evacuation scenarios found that robots using active "nudging" behavior (e.g., guiding users toward exits) resulted in faster, more efficient evacuations compared to passive "waiting" robots — particularly when users were unfamiliar with the environment [21]. This suggests that robots can shape human decision-making via behavior or interface design. Translating such strategies to teleoperation, informational nudges could be implemented via the control interface or haptic feedback — for instance through visual emphasis, suggestions, or predictive cues — to reduce operator errors, lower cognitive load, and promote safer or more efficient operator decisions.

2.2.3 Recommender Systems for Operator Support

Recommender systems [35] — common in domains like e-commerce or media — can be repurposed for teleoperation to provide context-aware, personalized suggestions: proposing feasible goals, trajectories, grasp targets, or control modes based on current robot state, environment, and inferred operator intent. A recommender module could rank candidate actions by predicted success, safety, or operator suitability, then present them to the operator in prioritized order.

Combined with nudging principles, such recommendations subtly steer operator behavior while preserving control freedom.

2.2.4 Ethical and Control Considerations

Integrating nudges or recommender guidance into teleoperation raises ethical and practical issues [27, 34, 43]. Excessive reliance on recommendations or overly directive nudges may reduce situational awareness, erode operator authority, or foster automation over-trust. Conversely, opaque recommendation logic or lack of transparency can lead to confusion or distrust. For safety-critical tasks, it is essential that assistance remains optional, interpretable, and overrideable. The design must emphasize transparency, user control, and adaptability — allowing operators to ignore suggestions or disable nudging if desired.

2.3 Human Decision-Making Models

2.3.1 Classical Decision Models

The classical approach to decision-making treats agents as rational actors operating under full information and unlimited cognitive resources. In this framework, individuals evaluate all available alternatives, assess their outcomes, and select the option that maximizes expected utility — effectively performing a cost–benefit analysis under uncertainty. The seminal work formalizing this is the John von Neumann–Oskar Morgenstern utility theory, which proves that if an agent's preferences satisfy certain axioms, there exists a utility function and the agent behaves optimally in the sense of maximizing expected utility [38]. Extensions to accommodate subjective beliefs under uncertainty, such as the framework developed by Leonard J. Savage (subjective expected utility), further formalize how agents might act rationally when probabilities are not objectively known but represented by subjective beliefs [49]. This theoretical foundation provides a normative baseline reflecting "ideal" or "optimal" operator behavior — useful when designing autonomous modules or defining optimal action/policy in robotics under idealized assumptions.

2.3.2 Bounded Rationality and Dual-Process Perspectives

In realistic settings — such as teleoperation — the assumptions of full information and unlimited computation rarely hold. Humans face cognitive constraints: limited memory, processing capacity, and time pressure. Herbert A. Simon argued that under such constraints agents adopt simplified decision rules rather than exhaustively optimizing — a concept known as *bounded rationality* [5]. Rather than maximizing utility over all alternatives, bounded-rational agents may "satisfice" — selecting the first option that meets some satisfactory threshold [5]. Heuristic decision-making (rules of thumb) emerges in these conditions: quick, low-cost mental shortcuts that trade off optimality for speed and cognitive economy [17]. More recent work frames these processes within a broader theoretical context, arguing that heuristics and procedural rationality can outperform fully rational models under realistic ecological and computational

constraints [14]. For teleoperation contexts — especially under time constraints, uncertainty, limited sensory feedback, or high cognitive load — bounded rationality combined with heuristic or dual-process thinking offers a more realistic model of operator decision behavior than classical utility maximization.

2.3.3 Implications for Computational and Learning-Based Models

Given that neither perfect rationality nor fixed heuristics fully capture human behavior in complex, dynamic environments, computational models often adopt probabilistic or learning-based frameworks. For instance, operator decision making can be modeled as probabilistic inference under uncertainty, where prior beliefs and sensory/contextual evidence combine to inform likely decisions [42, 53]. Such models allow for uncertainty, adaptation, and variability, aligning more closely with real human behavior under bounded rationality. Moreover, learning-based paradigms (e.g., reinforcement learning, adaptive control) can capture how operators adapt their decisions over repeated tasks — shifting heuristics, refining internal models, or improving performance through feedback. These frameworks are particularly useful in teleoperation systems where operators learn over time, and the system can adapt assistance or recommendation policies to individual operators' behavior patterns.

2.3.4 Modeling Operator Behavior in Teleoperation via Decision Theory

In teleoperation robotics, decision-theoretic models provide a principled basis for modeling operator behavior, anticipating choices, detecting suboptimal decisions (due to bounded rationality or cognitive load), and offering contextual support. Probabilistic modeling of operator intent can drive shared-control arbitration; recommender systems can suggest safer or more efficient actions; nudging mechanisms can guide operators toward high-utility or low-risk choices — all while acknowledging that operators may not, or cannot, perform full utility optimization. By combining classical normative models (as an ideal benchmark) with bounded-rationality–based descriptive models (as realistic behavior approximation) and computational inference models, one can build a robust theoretical foundation for a teleoperation system that supports human decision-making in a principled yet practical way.

3 System Architecture

3.1 Overview

The overall system is structured as a distributed teleoperation architecture with a clear separation between a local operator interface and a remote robotic avatar. The operator side comprises a VR-based visual and audio interface and a haptic device (Sigma 7) used for motion input and force feedback. The remote side comprises a Franka Emika Panda arm with a gripper and a pan–tilt unit carrying a stereo camera and microphones. Both sides are connected over a dedicated wired local network using a lightweight UDP-based middle ware. A central coordination process, running on the avatar PC, maintains global system state and mediates between the devices and the recommendation system.

Figure 3.1 illustrates the main components and data flows. Each physical device (haptic interface, arm, head/pan-tilt unit, avatar aggregation node) is implemented as an independent process with its own state machine and real-time control loop. The central controller monitors the status of all devices, manages state transitions, and serves as the integration point for perception, task-level reasoning, and the recommender.

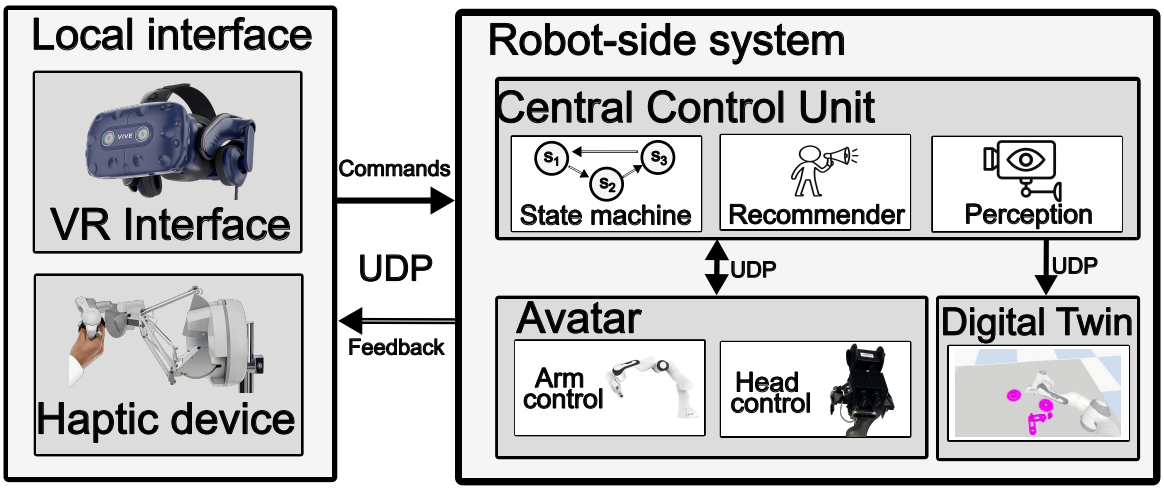


Figure 3.1: System overview and data flow. The operator interacts through a VR interface (HMD and controller) and a sigma.7 haptic device. Commands are transmitted to the robot-side system, where the central control unit coordinates the supervisor state machine, recommender, and perception. The avatar executes arm and head control and returns feedback to the operator. A digital twin mirrors the system state for visualization and debugging.

3.1.1 Thread structure and execution rates

All device processes follow the same three-thread pattern:

- **State thread** (*state handler*): a low-frequency thread that executes the device's internal state machine. It runs at $f_{\text{state}} = 100$ Hz and is responsible for mode transitions (offline, idle, standby, handshake, awaiting engagement, engaged, pause, home, error, stop). It enforces local safety logic, reacts to high-level commands from the central controller, and handles local fault conditions (e.g. loss of device communication, driver errors).
- **Data thread** (*data handler*): an intermediate-frequency thread that manages all incoming and outgoing messages. It reads the latest packets from the UDP communication layer, updates internal command variables, and publishes the current device state (poses, wrenches, twists, gripper status, faults) to the network. The update frequency depends on the device: for real-time controlled components such as the arm and haptic interface it is configured up to 800 Hz, whereas purely supervisory nodes (e.g. avatar aggregation) can run at lower rates.
- **Control thread** (*control handler*): a high-frequency thread that executes the actual control law on the hardware, typically at $f_{\text{control}} = 1$ kHz. For the haptic device this realizes Cartesian impedance and guidance force rendering; for the arm it implements Cartesian impedance control with nullspace posture regulation; for the pan–tilt head it drives the joint-level torque control. The control thread reads its setpoints and external signals from shared memory updated by the data thread, using mutexes to avoid race conditions.

This separation ensures that time-critical control loops remain deterministic and isolated from network delays and higher-level logic, while the state and data threads handle mode management and communication at lower computational cost. The choice of 100 Hz for the state thread is sufficient to react quickly to safety events and user commands without overloading the CPU, whereas the 1 kHz control loops satisfy the real-time requirements of haptic rendering and robot torque control.

3.1.2 UDP communication layer

Inter-process communication relies on UDP sockets, with messages encoded using MsgPack [36] to achieve compact, low-overhead binary serialization. Each device exposes one or more communication objects responsible for transmitting and receiving a shared message structure comprising the device type, current state, timestamp, and payload. The communication subsystem employs dedicated send and receive threads operating at fixed frequencies (up to 800 Hz) to prevent blocking. Because teleoperation primarily depends on the most recent state information, older packets are discarded to minimize latency [39]. In addition, the system continuously monitors incoming traffic to detect whether messages are still being received or whether communication has ceased, enabling timely identification of safety-related shutdowns.

3.1.3 Central coordination and global state

Global system coordination is implemented in Python and runs on the avatar computer, co-located with the robot and perception pipeline to minimize latency for the data-intensive streams. The central node maintains a global system state $s_t \in \mathcal{S}$ (offline, idle, standby, handshake, awaiting engagement, engaged, pause, home, stop) that is derived from the individual states and connectivity of the operator interface, avatar node, and haptic device:

$$s_t = f\Big(s_t^{\text{operator}}, s_t^{\text{avatar}}, s_t^{\text{haptic}}, c_t^{\text{operator}}, c_t^{\text{avatar}}, c_t^{\text{haptic}}\Big).$$

This logic is encapsulated in a dedicated state machine class running at 100 Hz, implementing a finite-state machine over $\mathcal{S}$. It enforces consistent transitions such as permitting `idle` → `standby`/`handshake` only when operator and avatar ar connected, entering `awaiting_engagement` only once the avatar is in the handshake pose, and reverting to `idle`/`standby` on user stop requests or loss of connectivity of any critical component.

The central state is broadcast continuously to all devices. Each local state machine interprets this as a command signal but retains authority to reject unsafe transitions (e.g., hardware faults) and to enforce device-specific safety behavior. This hierarchical scheme ensures that no single component failure or stale command can result in unsafe motion.

3.1.4 Integration of interface, avatar, and recommender

Within this architecture, the VR interface, haptic device, and avatar appear as peers coordinated by the central controller:

- The **operator interface** (VR) publishes user inputs and head pose, and renders video, audio, and HUD overlays driven by the central controller and the recommender.
- The **haptic device** streams handle pose and twist to the arm as control reference, and receives end effector pose, twist and estimated contact wrench as feedback signals.
- The **avatar node** acts as a high-level coordination layer that wraps the arm and head sub-components, handling their state changes, reacting to commands from the central controller, and propagating device-level faults or status updates.

Beyond the state machine and the communication layer, both the recommendation system and the perception module are integrated directly into the central control unit. This integration leverages the unit's complete access to the sensor state space, eliminating additional communication delays and avoiding potential effects of packet loss. The subsequent sections detail the design of the teleoperation interface, avatar control, perception pipeline, and recommender integration.

3.2 Teleoperation interface

The teleoperation interface is designed to make the remote robot avatar feel as natural and transparent as possible to the operator. A central goal is to create a strong sense of embodiment, allowing users—especially novices—to interact with the system without extensive training [13, 50]. This is crucial for the present work, which investigates how informational nudging

influences human decision-making; if the interface itself were difficult to use, usability issues would overshadow any effect of the recommender.

To achieve this, the interface integrates three tightly coupled channels: a VR-based visual interface providing a first-person view and head-coupled camera motion, a haptic interface that renders contact and assistance forces, and an audio channel delivering environmental sound and optional cues (Fig. 3.2). Their feedback is designed to be internally consistent and aligned with the robot's motion, ensuring that nudges from the recommendation system can be added without disrupting the operator's perception. The following subsections describe the visual and haptic components in more detail.

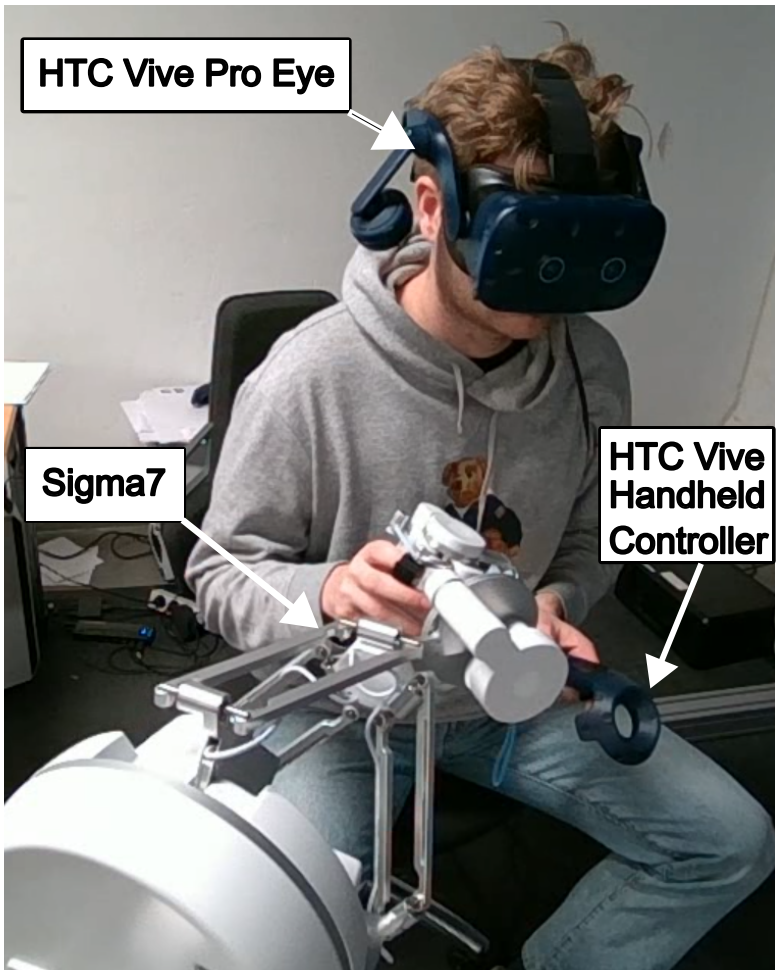


Figure 3.2: Operator interface during a trial, showing the VR head-mounted display (visual channel), the sigma.7 haptic device (force feedback and guidance), and the handheld controller (UI interaction).

3.2.1 Visual interface

The visual interface is implemented using an HTC Vive headset connected to an Unreal Engine 5.4 application written entirely in C++. A native C++ implementation avoids the additional abstraction layers of Blueprint scripts and provides tighter control over timing,

memory allocation, and integration with third-party modules. The VR application receives all robot and recommender data through the same UDP-based communication framework as the other devices and exposes a single VR Operator class responsible for state updates, UI logic, and user input.

The visual channel is designed to present a clear and unobtrusive representation of the robot's workspace, enabling operators to understand the scene quickly and act effectively. It renders a first-person view (FPV) from the robot-mounted camera as the primary background, overlaid with a head-locked HUD for system status and controls, and task-locked graphical elements that highlight the current target or provide local guidance near the robot wrist.

Low-latency video streaming The main camera stream is transported using the Network Device Interface (NDI) protocol over the wired local network and decoded directly inside Unreal. NDI provides a compressed video stream with very low end-to-end delay, which is critical for teleoperation comfort and stability in VR. Two camera types were evaluated: (i) a Kandao 8K VR180 stereo camera, which can deliver a stereoscopic FPV stream, and (ii) an Intel RealSense D455i depth camera used in monocular mode. While the Kandao camera offers higher spatial resolution and the potential for stereoscopic rendering, the necessary conversion from the camera's internal codec to NDI, in combination with the higher pixel throughput, led to increased GPU load and a significantly higher photon-to-eye latency on the target hardware. The RealSense camera, by contrast, provides more modest frame sizes but can be streamed directly as NDI, resulting in substantially lower latency and more stable frame rates. In the final system, the monocular RealSense stream was therefore preferred as it offered a better overall trade-off between image quality and teleoperation comfort.

Stereo layers and HUD design All interface elements are rendered via Unreal stereo layers that are head-locked to the VR camera. In contrast to placing 3D widgets on a plane in the scene, stereo layers bypass the standard rendering pipeline and are composited directly in the HMD, which reduces perceived latency and eliminates small head-motion–induced jitter that proved irritating in early prototypes. Although stereo layers are restricted to 2D textures, this constraint is acceptable for the HUD and enables per-eye textures if stereoscopic overlays are desired in future work.

The visual interface layout (Fig. 3.3) follows a clear semantic structure and separates information into *head-locked* and *world-locked* elements.

Head-locked information comprises persistent, operator-centric data that must remain continuously visible regardless of the user's gaze direction. This category includes:

- A lower control bar providing interactive teleoperation widgets (start, engage, home, nudge, clutch) as well as parameter adjustment elements for modifying control settings during operation.
- Status and monitoring displays located in the upper and lateral regions of the HUD, presenting global information such as end-effector height relative to the workspace, device engagement states, remaining task time, gripper status, and radial gauges visualizing force, torque, and speed.

World-locked information is spatially anchored to task-relevant objects in the environ-

ment—typically the robot arm or the target—and provides contextual cues for perception and action:

- Alignment bars, angle needles, and similar geometric guidance elements.
- Bounding boxes, textual annotations, or proposed movement paths.
- Wrist-mounted indicators for local force/torque feedback, grasp suggestions, or height cues.

This distinction reflects the functional difference between globally relevant system information and local task-relevant cues. Head-locked elements convey information that is not tied to the workspace but remains important throughout teleoperation (e.g., safety indicators, timing, global state). In contrast, world-locked elements are placed directly within the operator's attentional focus. Human operators naturally fixate on the region of interaction, and under high cognitive load tend to suppress peripheral visual content. To ensure that critical task-specific cues are reliably perceived, they are spatially collocated with the end-effector or target. This design leverages human attentional behavior to improve situational awareness and reduce cognitive demands during manipulation.

HUD Interaction. User interaction with the HUD is performed via a VR cursor coupled to the hand controller and a trigger button. The cursor motion is smoothed through temporal filtering and constrained to the HUD canvas. Button activations are detected using axis-aligned bounding-box (AABB) tests on the cursor's layer-space coordinates. Upon activation, interface widgets provide lightweight visual feedback (e.g., size or color modulation) as well as short auditory confirmation cues. This interaction mechanism enables reliable selection of HUD elements without diverting attention from the task space.

Point projection. To spatially align graphical overlays with real objects in the video stream, the system uses a calibrated projective camera model. Known 3D poses of the camera, pan–tilt unit, robot end-effector, and detected objects (e.g., via AprilTags) are first expressed in the camera coordinate frame through standard rigid-body transformations. These points are then projected into pixel coordinates using the camera's intrinsic parameters and distortion coefficients, obtained from a checkerboard calibration performed with OpenCV. This procedure provides an accurate world-to-image mapping and enables precise placement of augmented elements within the visual interface.

Digital twin view In addition to overlays in the camera image, the interface provides a compact digital twin of the workspace. A simplified PyBullet [11] model of the robot and objects is rendered on a small panel in the upper-right HUD region. The operator can toggle different virtual viewpoints (first-person, bird's-eye, frontal) via HUD buttons. During critical task phases, such as tight insertions, the recommender can temporarily promote the digital twin panel to a target-locked or wrist-locked position to provide a clearer geometric view; after a short timeout it returns to its resting HUD position to avoid distraction.

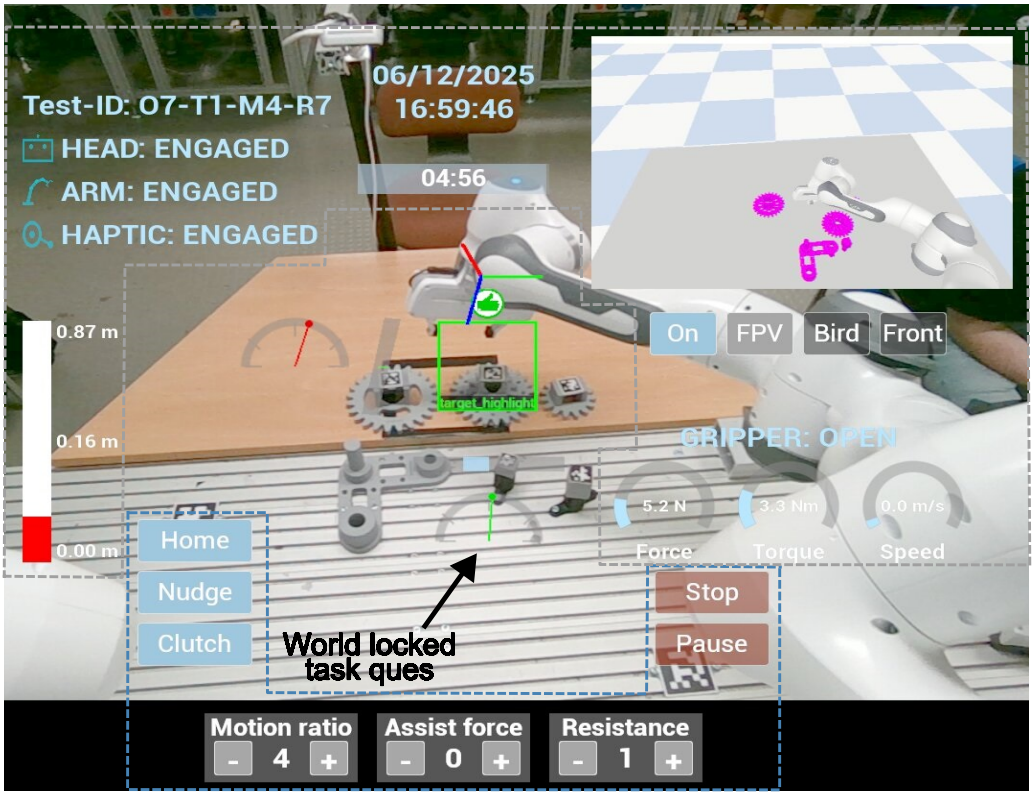


Figure 3.3: Operator view with layered visual augmentation. Screen-fixed controls are placed at the lower edge, while persistent system state information remains visible along the periphery. Task-relevant cues are overlaid directly in the workspace as world-locked annotations, complemented by robot-locked reference frames attached to the manipulator.

3.2.2 Audio interface

The audio interface complements the visual and haptic channels by reproducing the acoustic scene of the remote workspace and by conveying additional cues. To approximate what an on-site operator would hear, a Free Space Pro II binaural microphone is rigidly mounted behind the camera on the pan–tilt unit. The two microphone channels are streamed over the local network using the same NDI-based pipeline as for video and are rendered through the headphones of the VR headset. This preserves basic spatial cues and head–camera synchrony, contributing to the sense of presence.

In addition to this task-space audio, the Unreal application can play short sound effects for confirmations, encouragements, and warnings. These are mapped to specific events (e.g. successful grasp, timeout, force limit exceeded) and are kept brief to avoid masking environmental sounds.

Finally, a text-to-speech module is available to provide spoken messages to the operator. This channel is particularly useful for conveying discrete instructions or explanations (e.g. “the selected part is not admissible”) without requiring the operator to divert gaze from the task or interpret additional symbols on the HUD, thereby keeping the cognitive load of information presentation low.

3.2.3 Haptic interface

The haptic interface is realized with a Force Dimension Sigma 7 [15] device operated in gravity-compensated mode, so that the controller only needs to generate interaction forces. Its main purpose is to convey what the robot "feels" to the operator and to provide additional information in fine-grained contact situations where visual feedback is limited by self-occlusion or camera resolution.

The mapping between handle motion and robot motion is defined by an origin pose in the handle frame. During operation, the current handle pose is measured, its deflection from the origin is computed, and this deflection is mapped into the robot base frame via a fixed rigid transform and a workspace scaling factor. The resulting SE(3) command is sent as the reference pose for the Cartesian impedance controller of the arm. Conversely, the arm sends back its current end-effector pose, twist, and an estimate of the external contact wrench, which are used to render forces at the handle.

The total wrench rendered to the operator is composed of four conceptually distinct components:

1. **Teleoperation impedance.** A low-stiffness virtual coupling between the handle and the end-effector pose (mapped into the Sigma 7 frame) creates a sense of inertia and pose feedback and damps sudden movements. For passivity and stability, damping is applied with respect to the measured handle velocity rather than the arm velocity. The stiffness and damping of this coupling are tuned such that feedback is informative but does not tire or overpower the operator.
2. **Contact wrench feedback.** The external wrench estimated at the robot end-effector is transformed into the haptic device frame and rendered as a contact force component. This is particularly important for the insertion task, where subtle changes in contact forces and torques indicate misalignment. Rate limits and magnitude bounds on the rendered wrench ensure stable and comfortable feedback.
3. **Guidance (assistance) force.** When the recommender provides a current target pose, an additional virtual guidance field attracts the handle towards the corresponding position and orientation. This guidance wrench is derived from a non-linear potential field that smoothly increases with distance and orientation error and is scaled by a context-dependent assistance gain. It reduces operator effort and helps to stabilize motion towards the intended goal while preserving the ability to deviate or override the guidance.
4. **Insertion nudging wrench.** For high-precision insertions, an extra nudge component is activated that depends only on the estimated contact wrench, not on the exact nominal target pose. It analyses the dominant torque components during contact and generates a corrective torque at the handle that tends to reduce these components, effectively steering the end-effector towards a configuration with lower contact moments. Because this nudge is purely force-based, it remains helpful even if the exact insertion position is uncertain or slightly miscalibrated.

All components are summed to form a desired task-space wrench, which is then subject to rate limiting and clamping to device-specific bounds before being applied at the handle. This ensures passive, stable interaction and keeps forces within comfortable limits while still conveying rich information about contact events, assistance, and insertion guidance to the operator.

3.3 Robot Avatar System

The remote robot avatar is realized on an existing teleoperation platform available in the lab. It consists of a Franka Emika Panda arm mounted on a vertical wall with a mounted two finger gripper and a pan–tilt head unit carrying the main camera and binaural microphone. The arm is positioned on the operator's right-hand side beneath the head module, and a table in front of the platform defines the main workspace for all tasks. This arrangement mimics a human upper body: the camera and microphone approximately coincide with a head position above the "shoulders", while the arm reaches into the workspace in front of the torso. The pan–tilt unit provides two rotational degrees of freedom so that the avatar can look left–right and up–down, closely matching natural head movements.

Figure 3.4 shows the overall physical setup together with the world, robot-base, and camera frames. A separate perception camera with calibration tags on the table is used to estimate object poses in the workspace and to anchor task-specific coordinate frames. These calibrated frames form the geometric backbone for the mappings used by the teleoperation interface, perception, and recommender, and will be referred to throughout the remainder of this chapter.

3.3.1 Simulation environment

To support safe and rapid development, a physics-based simulation of the avatar platform was implemented using the physics simulator MuJoCo [52]. The simulator reproduces the kinematic layout of the real setup: a Franka Emika Panda arm and a model of the real pan–tilt unit are mounted in the same configuration as on the physical platform, and the main camera is rigidly attached to the virtual pan–tilt module.

The control software is written such that it can operate either on the real hardware or on the simulator without modification. A custom `franka::Robot` wrapper forwards the same Cartesian impedance control callback used on the real robot to the MuJoCo model, and exposes compatible interfaces for reading the robot state, Jacobian, and mass matrix. At compile time, a configuration flag selects whether the control loop connects to the physical robot drivers or to the simulator running on the local PC. From the perspective of the higher-level architecture (central controller, teleoperation interface, recommender), the simulated robot is therefore indistinguishable from the real one.

This setup allows algorithms and control changes to be tested first in simulation, including contact-rich interactions, without risking hardware damage. It also enables repeatable experiments and debugging of corner cases while keeping the control structure and timing behavior as close as possible to the real system, thus reducing the sim-to-real gap.

3.3.2 Arm control

The avatar arm is controlled in Cartesian space using an impedance control scheme with a simple null-space controller for posture regulation. The choice of Cartesian impedance over classical joint-space control is motivated by two considerations. First, it is computationally lightweight and can be executed in a 1 kHz torque loop without pre-computing trajectories

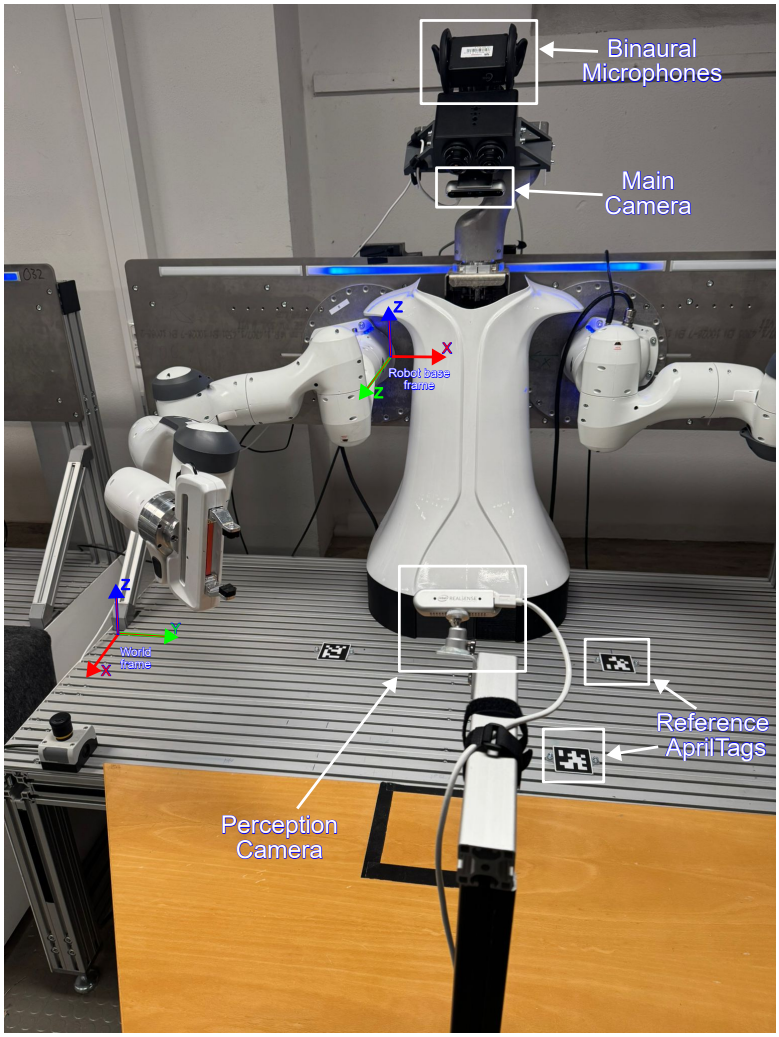


Figure 3.4: Experimental teleoperation setup showing the robot arm, end-effector, and task workspace. Coordinate frames of the robot base, arm, and end-effector are indicated for reference.

or solving optimization problems online. Second, it directly regulates end-effector motion in task space, which is where the operator perceives and reasons about the task (positions and orientations of the gripper relative to objects). In contrast, joint-space controllers operate on individual joint angles; mapping operator commands to suitable joint trajectories would require an additional inverse-kinematics and trajectory generation layer, and the resulting joint motion is harder for the operator to interpret and predict from the visual feedback. Cartesian impedance control also provides a convenient way to specify task-space compliance and contact behavior, which are central in the present teleoperation setting [19].

Cartesian impedance Let $x \in SE(3)$ denote the current end-effector pose in task space and $x_d \in SE(3)$ the desired pose generated from the teleoperation mapping. For control, we work with a minimal 6D pose error $e(x_d, x) \in \mathbb{R}^6$ obtained from the logarithm map on $SE(3)$, combining position and orientation error. The desired pose x_d is updated continuously from the haptic handle via a fixed rigid transform and a workspace scaling factor. To avoid large torque

transients when the reference changes significantly (e.g. at the start of a task or when switching targets), a simple linear interpolation in task space is applied between the previous and new reference pose, limited by a maximum admissible twist magnitude. Because these commands are updated at high rate and Cartesian impedance is robust to moderate reference steps, no explicit time-parameterized trajectory generation is required.

The Cartesian impedance law is defined as

$$F_{\text{task}} = K\, e(x_d, x) + D\Big(\dot{x}_d - \dot{x}\Big), \tag{3.1}$$

where $K, D \in \mathbb{R}^{6\times 6}$ are positive semi-definite stiffness and damping matrices, and $\dot{x}, \dot{x}_d$ denote the current and desired task-space velocities. In the present implementation, the desired twist is set to zero, $\dot{x}_d = 0$, so that the controller attempts to bring the end-effector to the desired pose and then hold it there. This choice avoids "chasing" noisy or delayed command velocities from the teleoperation channel and contributes to passive, stable behavior: the human moves the handle to update x_d, but once the handle is stationary the robot settles to a static equilibrium.

Stiffness and damping are chosen isotopically within translation and rotation,

$$K = \operatorname{diag}(k_p^{\text{lin}} I_3,\ k_p^{\text{rot}} I_3), \qquad D = \operatorname{diag}(k_d^{\text{lin}} I_3,\ k_d^{\text{rot}} I_3), \tag{3.2}$$

with scalar gains $k_p^{\text{lin}}, k_p^{\text{rot}}, k_d^{\text{lin}}, k_d^{\text{rot}} > 0$. In free motion, this controller makes the end-effector behave like a virtual mass–spring–damper system that tracks the desired pose with compliant behavior, perceived by the operator as slightly damped but responsive arm motion. The gains are tuned to balance tracking accuracy and compliance.

Given the robot Jacobian $J(q) \in \mathbb{R}^{6\times 7}$ at the current joint configuration q, the corresponding joint torques are

$$\tau_{\text{task}} = J(q)^\top F_{\text{task}}. \tag{3.3}$$

This command is combined with a model-based gravity compensation term $\tau_{\text{grav}}(q)$ provided by the robot model so that the impedance behavior is largely independent of the arm configuration.

Null-space posture control The operator does not directly command the redundant degrees of freedom of the 7-DoF arm. Without additional regulation, the elbow could drift into configurations that interfere with the environment or are visually confusing, even if the end-effector tracks the desired pose correctly. To avoid this, a simple null-space controller is added that keeps the arm close to a comfortable nominal posture q_0 while minimally influencing the task-space behavior.

Let $N(q) = I - J(q)^\top \Big(J(q)J(q)^\top\Big)^{-1} J(q)$ denote a projector onto the null-space of the Jacobian. A quadratic posture potential

$$V_{\text{null}}(q) = \frac{1}{2}(q - q_0)^\top K_{\text{null}}(q - q_0) \tag{3.4}$$

with a diagonal, anisotropic gain matrix $K_{\text{null}} \succeq 0$ yields the joint-space control law

$$\tau_{\text{null}} = -N(q) K_{\text{null}}(q - q_0). \tag{3.5}$$

The gains are chosen such that posture regulation primarily acts on shoulder and elbow joints, while the wrist joints are weighted weakly. This keeps the global arm posture stable without

noticeably constraining wrist and hand motion, which remain free and dexterous for task execution.

The nominal posture q_0 is chosen such that the elbow remains roughly at the side of the torso and does not intrude into the main workspace in front of the table. Because τ_{null} is projected into the nullspace of $J(q)$, it does not generate forces in task space and therefore does not leak into the end-effector behavior perceived by the operator.

Resulting control law and implementation The overall commanded joint torque is

$$\tau = \tau_{\text{task}} + \tau_{\text{null}} + \tau_{\text{grav}}(q), \tag{3.6}$$

with additional saturation and rate limiting to enforce torque and velocity limits. The controller is implemented in C++ using the `libfranka` [16] torque-control interface. A dedicated real-time loop at 1 kHz evaluates the robot state, computes the Jacobian and model terms via the library's model API, updates the impedance and null-space torques, and sends the result to the robot.

In summary, Cartesian impedance control with null-space posture regulation provides a natural and robust mapping from operator commands to avatar motion. It offers intuitive behavior in task space and can be executed in real time without precomputed trajectories. At the same time, its inherently compliant nature makes it well suited for contact-rich teleoperation and stable haptic feedback, even under moderate modeling errors and variable user input.

3.3.3 Head control

The head of the avatar is realized as a two-DoF pan–tilt unit that follows the operator's head orientation. From the VR system, the yaw and pitch angles of the HMD are extracted and mapped to desired pan and tilt angles for the head joints. A small constant offset in pitch is applied so that the avatar naturally looks slightly downward, reducing the effort required from the operator's neck while maintaining an intuitive mapping.

Head motion is controlled in joint space using an impedance-like position controller. Let $q \in \mathbb{R}^2$ denote the current pan–tilt joint angles and $q_d \in \mathbb{R}^2$ the desired angles derived from the HMD. A simple path planner interpolates between the current and desired angles so that large changes (e.g. at engagement or after a pause) are executed smoothly and bounded by a maximum angular velocity. The torque command is given by

$$\tau_{\text{head}} = K(q_d - q) - D\dot{q}, \tag{3.7}$$

where $K, D \in \mathbb{R}^{2\times 2}$ are diagonal stiffness and damping matrices, and $\dot{q}$ is the calculated joint velocity using forward Euler and small low-pass filter. The damping term uses the head's own joint velocities, which stabilizes the behavior even in the presence of communication jitter or small delays in the HMD signal. Stiffness is adapted as a function of the tracking error to avoid overly aggressive torques for large deviations. Joint angles and torques are limited, and pan and tilt are constrained to lie within predefined bounds so that the head cannot move into uncomfortable or unsafe configurations for the operator.

The head controller runs in a real-time torque loop at 1 kHz on the avatar PC, while a slower data thread (at 200 Hz) exchanges the current joint angles and system state with the central

controller and the VR interface. The camera and binaural microphone used for the visual and audio interface are rigidly mounted on the tilt frame. As a result, changes in the operator's head orientation directly and with low latency cause corresponding rotations of the remote camera and microphone. The same joint angles are used in the projection pipeline of the visual interface to map world-space objects into image coordinates. This tight coupling between head motion, visual perspective, and audio scene is essential for maintaining a strong sense of presence and for avoiding motion sickness during teleoperation [9, 23, 26, 48, 54].

3.4 Perception and world representation

3.4.1 Object detection and tracking

To obtain object poses in the workspace, a dedicated overhead perception camera observes AprilTags [41] attached to all relevant parts. In the current setup, a color camera (Intel RealSense D435i) is mounted above the table and streams RGB images at 1280×720 px at 15 FPS. The frame rate is sufficient for the quasi-static object motions considered here and keeps computational load moderate.

For each image, an AprilTag detector estimates the pose of all visible tags in the camera frame using a calibrated pinhole model. A small set of larger reference tags are placed on the workspace table, while smaller tags are attached to individual objects. The reference tags are used to initialize and stabilize the workspace geometry; the remaining tags provide per-object poses.

To reduce jitter and spurious updates, the raw poses are low-pass filtered in position and orientation, and changes below small thresholds are ignored. The perception module exposes, for each tracked object ID, a homogeneous transform in SE(3) together with a timestamp, so that the central controller and recommender can access object positions, orientations, and detection freshness. This design allows the system to react to changes in object placement online without relying on a fixed, pre-programmed arrangement and, in principle, extends to multiple cameras if higher accuracy or coverage is required.

3.4.2 World frame, task frames, and calibration

All geometric quantities in the system are expressed relative to a common world frame $\{W\}$ defined on the table in front of the robot. The origin and axes of $\{W\}$ are chosen once during setup (in the present case near the front–right corner from the robot's perspective) and then kept fixed; their exact location is arbitrary as long as the choice is consistent.

The perception camera is related to $\{W\}$ by a rigid transform obtained from an offline calibration procedure. In practice, the pose of a designated reference tag on the calibration board is measured in the world frame, and the camera-to-tag transform is recovered from images using standard camera calibration and pose estimation [7, 56]. Composing these transforms yields the camera pose in $\{W\}$ and allows all detected object tags to be converted into world-frame poses.

The robot base frame $\{B\}$ is likewise linked to the world frame by a fixed transform determined during installation. This enables straightforward conversion between end-effector poses from the

arm controller and the object poses from perception. The head camera used for teleoperation is mounted on the pan–tilt unit with a known transform relative to the joints; combining this with the current head angles yields its pose in $\{W\}$ and underpins the world-to-image projection used for HUD overlays.

Task-specific frames, such as fixtures or assembly bases, are defined as additional rigid transforms relative to $\{W\}$ or to individual object tags. In summary, the perception and calibration pipeline provides a consistent SE(3) representation of the workspace in a single world frame, so that teleoperation control, visual overlays, and the recommendation system all operate on the same geometric backbone.

3.5 Central coordination and communication

The central controller is responsible for coordinating all devices, managing the global system state, and integrating the recommendation system. It runs on the avatar PC and interfaces with the operator VR application, haptic interface and the avatar process via the UDP-based communication infrastructure described in Section 3.1. Perception is implemented as a local module on the same machine and is accessed directly via function calls. On top of this transport layer, the central controller implements the distributed state-machine logic and the periodic evaluation of nudging actions.

3.5.1 Distributed state machines

Each device (operator interface, haptic node, avatar/robot node) maintains its own local state machine that encodes device-specific modes such as `offline`, `idle`, `standby`, `handshake`, `awaiting_engagement`, `engaged`, `pause`, `home`, and `stop`. These local machines enforce hardware- and device-level safety constraints (e.g. watchdog timeouts, driver errors, emergency stop) and ensure that a device enters a safe configuration if communication is lost. Local states and connectivity flags are sent periodically to the central controller via the communication manager.

On the central side, a separate state machine aggregates the local information into a global system state

$$s_t \in \mathcal{S} = \{\texttt{offline}, \texttt{idle}, \texttt{standby}, \texttt{handshake}, \texttt{awaiting\_engagement}, \texttt{engaged}, \texttt{pause}, \texttt{home}, \texttt{stop}\},$$

which is updated at $f_{\text{exec}} = 100$ Hz in a dedicated thread. Denoting by $s_t^{\text{op}}, s_t^{\text{av}}, s_t^{\text{ha}}$ the local states of operator, avatar, and haptic device, and by $c_t^{\text{op}}, c_t^{\text{av}}, c_t^{\text{ha}} \in \{0, 1\}$ their connectivity flags, the central update can be viewed as a deterministic transition function

$$s_t = F\Big(s_{t-1},\, s_t^{\text{op}}, s_t^{\text{av}}, s_t^{\text{ha}},\, c_t^{\text{op}}, c_t^{\text{av}}, c_t^{\text{ha}}\Big), \tag{3.8}$$

implemented as a finite-state machine with explicit guards and illustrated in Fig. 3.5.

State semantics. For clarity, the supervisor states are interpreted as follows:

- `offline`: system not running and no device control active.

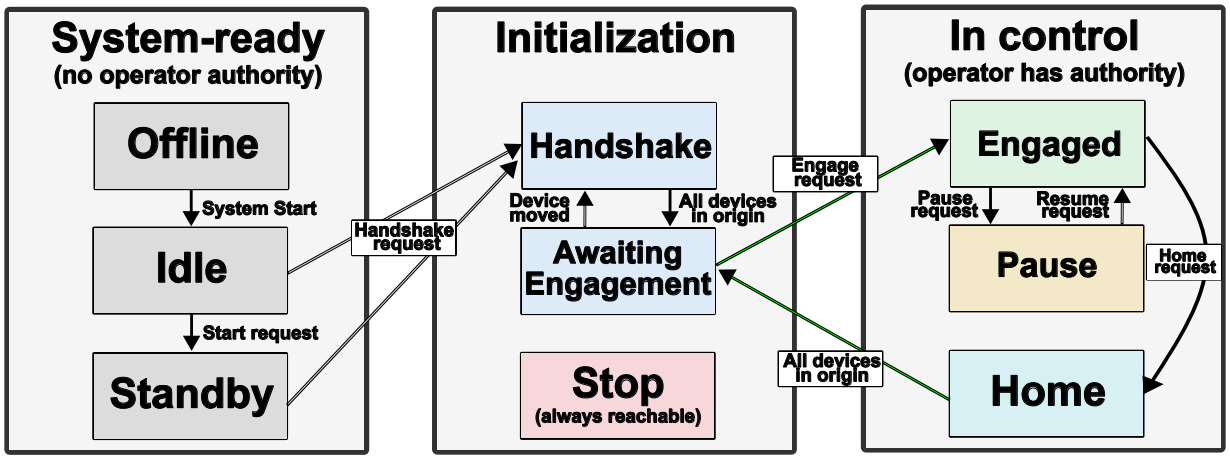


Figure 3.5: Distributed teleoperation supervisor state machine. States are grouped into system-ready (no operator authority), initialization, and operator-control phases. During initialization, `handshake` aligns device reference frames and moves all components to a consistent origin pose before entering `awaiting_engagement` and enabling `engaged`. A global `stop` state (always reachable) provides an immediate safety override from any state.

- `idle`: system running and listening for start requests; no control output.
- `standby`: control loops active in a safe holding configuration; teleoperation not enabled.
- `handshake`: initialization that aligns device reference frames and moves devices to predefined origin poses.
- `awaiting_engagement`: all devices are in origin; the operator may engage teleoperation. Deviations or dropouts return to `standby` or `handshake`.
- `engaged`: operator has authority and commands are executed.
- `pause`: teleoperation suspended while maintaining a safe hold; can resume to `engaged`.
- `home`: autonomous motion to a predefined safe home pose; transitions to `awaiting_engagement` once reached.
- `stop`: global safety override that stops motion immediately and requires explicit reset before returning to `standby`.

The resulting global state s_t is continuously broadcast to all devices via the communication manager, so that each local state machine can interpret it as a requested mode. Devices are free to reject unsafe transitions, in which case the central state machine will settle back to a compatible state on the next update. This hierarchical design separates responsibilities: local state machines guarantee device-level safety and manage hardware interaction, while the central state machine coordinates experiment flow and participant interaction at the system level.

3.5.2 Integration of the recommender

The recommendation system is integrated directly into the central controller. This component runs in two background threads: one for evaluating potential nudges at a frequency of $f_{\text{nudge}} =$

1 Hz and one for distributing active nudges and updating task-level logic at a higher frequency $f_{\text{dist}} = 10$ Hz. The teleoperation control loops on the devices are unaffected by these threads and continue to run at their respective real-time rates.

At each execution step of the central controller, all incoming messages from the devices are collected by the communication manager and stored in a shared data structure. This aggregated state is passed to a nudge manager, which extracts a compact task state vector describing, among others, the current end-effector pose in the world frame, the relative error to the current target (if any), the handle wrench and twist, contact forces, grasp state, and intent-related quantities such as the predicted next part and its admissibility.

The nudging logic is gated by the global system state and task mode. In particular, recommendations are evaluated only during active task execution and are disabled in `idle`, `standby`, and `home` modes. At each nudge evaluation step, the nudge manager:

1. updates the task coordinator and intent inference with the latest data, including object detections and grasp state;
2. obtains the current target object and planned sequence from the planner;
3. constructs the context features required by the bandit-based recommender (Section **??**);
4. invokes the recommender to obtain sets of candidate *dynamic nudges* and *informational nudges*.

Candidate nudges are translated into a set of active dynamic and informational actions with associated time stamps and durations. A separate distribution thread manages the lifetime of these actions, removes expired nudges, and converts the remaining ones into outgoing messages. Dynamic nudges are encoded as parameter updates that are sent to the operator interface and, via it, to the haptic controller. Informational nudges are converted into modality-specific payloads and are transmitted to the VR interface.

Formally, the nudge manager implements a mapping

$$a_t = \pi_\theta(z_t),$$

where z_t is the current task and context state extracted from the aggregated data, and a_t is a structured action consisting of dynamic and informational components. The parameters θ of π_θ are updated online by the bandit algorithm based on observed rewards, while the central controller enforces safety and mode constraints by restricting when π_θ may be applied and by clamping all dynamic parameters to pre-defined safe ranges.

By co-locating the recommender with the central controller, all necessary information (teleoperation state, perception, task progress, and user parameters) is available without additional communication delays, and logging of context, actions, and outcomes for the user study can be performed consistently. At the same time, the separation between the real-time control loops on the devices and the slower, task-level recommender logic ensures that experimental nudging strategies cannot compromise basic teleoperation stability.

4 Nudging Recommender System

The recommendation system constitutes as the core contribution of this thesis. Its purpose is to influence a human operator's decisions during teleoperation by providing targeted feedback and assistance, while preserving the operator's overall authority over the robot. Instead of taking over control, the system aims to *nudge* the operator towards actions that are efficient, safe, and consistent with the task structure. In this sense, the recommender acts as an additional decision-making layer that observes the current situation, infers the operator's intent, and selects suitable informational and dynamic cues.

From a control perspective, the recommendation problem can be viewed as a mapping from a context

$$x_t \in \mathcal{X},$$

describing the current task state, robot state, and inferred operator intent, to a nudge

$$a_t \in \mathcal{A},$$

which modifies the feedback presented to the operator. The goal is to choose a_t such that a task-dependent performance criterion is improved in expectation, while constraints on transparency, user comfort, and interface limitations are respected. Formally, the recommender should maximize an expected cumulative reward

$$\mathbb{E}\left[\sum_{t=0}^{T} r(x_t, a_t)\right],$$

under the dynamics induced by the human–robot interaction and the teleoperation setup. The central design challenge is that the human component of these dynamics is only partially known, time-varying, and influenced by cognitive factors that are difficult to model explicitly.

The development of the recommendation system in this work followed two steps. In a first stage, a mathematically more elegant Stackelberg-style approach was implemented, in which the recommender acts as a leader that anticipates the response of a modeled human follower. The follower was represented by a stochastic optimal control model with belief dynamics, allowing the leader to compute nudges that are optimal with respect to a specified cost function. While this formulation is theoretically appealing and was realized in a full implementation, it proved difficult to align with actual operator goals, interface constraints, and the variability of human behavior.

In a second stage, these insights motivated a more pragmatic formulation based on contextual bandits. The final recommendation system decomposes the problem into two coupled components: a dynamic assistance bandit that adapts haptic and dynamic parameters of the teleoperation interface, and an informational bandit that selects visual and other feedback cues. Both bandits

operate on a context that includes inferred operator intent, task progress, and measurements from the teleoperation system, and they learn or exploit policies that are directly tied to observed task performance.

This chapter introduces the recommendation system in detail. First, the original Stackelberg-inspired formulation and its limitations are discussed, in order to clarify the modelling assumptions and the reasons for moving towards a data-driven approach. Subsequently, the contextual bandit formulation is presented, including the separation into dynamic and informational nudging, the construction of the context and action spaces, and the decision logic used in the experiments. Finally, the integration of the recommender with the teleoperation system, the intent inference, and the task-level planner is described.

4.1 Original Stackelberg Concept

In the initial design of the recommendation system, the operator and the teleoperation channel were modelled explicitly as a stochastic optimal control problem of LQG type. The recommender acted as a leader that selects feedback and assistance parameters, while the human operator was represented by a follower controller that optimizes a task-dependent cost based on a noisy internal belief of the state. This section first introduces the Stackelberg formulation and then details the follower and leader models used in this framework.

4.1.1 Stackelberg formulation

The overall interaction between recommender and operator is naturally described by a hierarchical, or leader–follower, structure. The recommender has access to task-level information (e.g. sensor data, admissible next parts, optimal assembly sequences, safety constraints) and can modify interface and assistance parameters, but it does not directly command robot motion. The human operator, in contrast, continuously controls the teleoperation handle and thus the robot motion, but only observes the state through the rendered visual, audio and haptic feedback. This asymmetry motivates a Stackelberg formulation: the recommender (leader) chooses a nudging strategy, anticipating the reaction of a modelled human follower, while the follower optimizes its own objective given the provided information and dynamics.

Let $\theta_t \in \Theta$ denote the vector of parameters chosen by the recommender at time t. These parameters include, for example, stiffness and damping values of guidance fields, gains for resistive or assistive forces, and switches that enable or disable certain information cues in the user interface. For a given parameter trajectory $\theta_{0:T} = (\theta_0, \dots, \theta_T)$, the follower applies a control sequence $u_{0:T}$ by solving a stochastic optimal control problem of LQG type based on its internal belief over the state. Denoting by $J_\mathrm{f}(x_0, \theta_{0:T}, u_{0:T})$ the follower cost functional and by $\mathcal{D}$ the teleoperation dynamics (including the EKF belief update), the follower's best response is

$$u^\star_{0:T}(\theta_{0:T}) \in \arg\min_{u_{0:T}} J_\mathrm{f}(x_0, \theta_{0:T}, u_{0:T}) \quad \text{s.t.} \quad (x_t, b_t) \in \mathcal{D}(x_{t-1}, b_{t-1}, u_{t-1}, \theta_{t-1}), \tag{4.1}$$

where x_t denotes the true teleoperation state and b_t the follower belief.

The recommender anticipates this best response and chooses $\theta_{0:T}$ to optimize its own objective J_l, which reflects task performance and additional design preferences such as smoothness, safety

margins, and limits on information load. In an idealized infinite-horizon setting, the leader problem can be written as the bilevel optimization

$$\theta^{\star}_{0:T} \in \arg\min_{\theta_{0:T}\in\Theta^{T+1}} J_{\mathrm{l}}\Big(x_0, \theta_{0:T}, u^{\star}_{0:T}(\theta_{0:T})\Big), \tag{4.2}$$

subject to the follower optimality condition (4.1) and additional constraints on θ_t, such as bounds on assistance gains and budgets for informational cues. This Stackelberg formulation formalizes the idea that the recommender does not directly override human control, but rather selects parameters of the feedback and assistance channels while anticipating how a rational follower, modelled here as an LQG controller, will react.

In the present work, this bi-level problem is not solved in closed form. Instead, it is approximated in a receding-horizon fashion: at discrete decision times separated by a nudging interval $\Delta t_{\mathrm{nudge}}$ the recommender treats the current state and belief as initial condition, simulates the follower model forward under a set of candidate parameter choices, and selects the parameter update that yields the largest predicted improvement in its objective over a finite horizon.

4.1.2 Follower model

The follower model specifies how the operator is assumed to behave for a given parameter trajectory $\theta_{0:T}$. It consists of three components: a dynamic model of the coupled teleoperation system, a belief estimator based on an extended Kalman filter (EKF), and a phase-dependent LQG controller that maps beliefs to control wrenches.

Teleoperation dynamics

The teleoperation system can be approximated by two coupled spring–damper systems. The operator interacts with the handle of the haptic device, which is modeled as a compliant interface. The robot arm is controlled by a Cartesian impedance controller which realizes another virtual spring–damper system at the end-effector. Both systems are coupled via a virtual spring and damper that transmit motion and force through the communication channel, which is subject to latency.

Let $x_h \in \mathbb{R}^n$ denote the pose of the haptic handle in its local frame, and $x_r \in \mathbb{R}^n$ the pose of the robot end-effector in the robot base frame. A fixed transformation T_{BH} maps handle poses into the base frame of the robot,

$$x_r^{\mathrm{cmd}} = f(T_{BH}, x_h),$$

where x_r^{cmd} is the commanded end-effector pose for the impedance controller. In the real system, x_r^{cmd} is transmitted via UDP and subject to a communication delay $\tau_c > 0$. Similarly, the measured robot pose x_r is sent back as a feedback signal to the haptic device side with delay.

On the device side, the handle dynamics are modeled as

$$M_h\ddot{x}_h + D_h\dot{x}_h + K_h(x_h - x_v) = u_h, \tag{4.3}$$

where M_h, D_h, K_h denote the positive semi-definite inertia, damping, and stiffness matrices of the haptic device, x_v is the virtual coupling state, and u_h is the control wrench exerted by the

human on the handle. The damping term $D_h\dot{x}_h$ is applied against the actual handle velocity, rather than a commanded velocity, as in the real implementation for stability reasons.

On the robot side, the Cartesian impedance controller is modeled as

$$M_r\ddot{x}_r + D_r\dot{x}_r + K_r(x_r - x_r^{\mathrm{cmd}}) = 0, \tag{4.4}$$

with M_r, D_r, K_r denoting the apparent inertia, damping, and stiffness of the impedance behavior. The commanded pose x_r^{cmd} tracks the delayed handle pose, i.e.,

$$x_r^{\mathrm{cmd}}(t) = f(T_{BH}, x_h(t - \tau_c)).$$

The force rendered to the operator is computed from the virtual coupling between x_h and the delayed robot pose $x_r(t - \tau_c)$, using the same spring–damper structure as in the real system. This yields a bidirectional coupling where the human wrench u_h excites the coupled system and the device renders the reaction forces corresponding to the remote environment.

Equations (4.3) and (4.4) can be combined into a state-space model

$$x_{t+1} = f_x(x_t, u_t) + w_t, \tag{4.5}$$

where x_t stacks handle and robot states (positions and velocities), u_t represents the human input, and w_t is process noise capturing modeling errors, unmodeled dynamics, and communication jitter.

Noisy observations and belief estimation

The human operator does not have direct access to the full state x_t. Instead, they perceive the system through visual, haptic, and auditory feedback. In the model, this is represented by a noisy observation process

$$y_t = h(x_t) + v_t, \tag{4.6}$$

where y_t collects visual measurements (end-effector pose in the rendered view, distances to objects, contact state) and haptic measurements (forces felt at the handle), and v_t is observation noise.

Different feedback channels are affected by different noise levels. For instance, visual distance estimates to task-relevant objects become more reliable close to contact but are coarse when the end-effector is far away. Haptic measurements are accurate when the handle is in contact with stiff surfaces but less informative in free space. To capture these effects, the observation model $h(\cdot)$ and the covariance of v_t are made context-dependent.

The operator's internal knowledge about the state is described by a belief

$$b_t = (\mu_t, \Sigma_t),$$

consisting of a mean μ_t and covariance Σ_t over x_t. Under the above modeling assumptions, the belief dynamics are approximated by an extended Kalman filter. Given the previous belief b_{t-1},

the applied input u_{t-1}, and the new observation y_t, the EKF prediction and update steps yield

$$\mu_{t|t-1} = f_x(\mu_{t-1}, u_{t-1}), \tag{4.7}$$
$$\Sigma_{t|t-1} = A_t \Sigma_{t-1} A_t^\top + Q_t, \tag{4.8}$$
$$\mu_t = \mu_{t|t-1} + K_t\Big(y_t - h(\mu_{t|t-1})\Big), \tag{4.9}$$
$$\Sigma_t = (I - K_t C_t)\Sigma_{t|t-1}, \tag{4.10}$$

where A_t and C_t denote the Jacobians of f_x and h at μ_{t-1} and $\mu_{t|t-1}$, Q_t is the process noise covariance, and K_t is the Kalman gain. The covariance Σ_t describes the operator's uncertainty about the state and plays a central role in how informational nudges are modeled.

Conceptually, the recommendation system can influence the observation model and thus the belief state. For example, providing an explicit numerical distance to a target object corresponds to a highly informative observation of that distance component, which reduces its variance and corrects the mean. If this information is no longer displayed, the uncertainty about the distance increases again over time through the prediction step. In the LQG follower model described below, the control law operates on the belief (μ_t, Σ_t), such that changes in information directly influence the predicted behavior.

Phase-dependent LQG follower

Human behavior during a teleoperated task is not stationary. Different phases of the task, such as approaching an object, aligning for insertion, or retracting, are associated with different priorities and strategies. To capture this, the follower controller is modeled as a phase-dependent LQG regulator.

Let $p_t \in \mathcal{P}$ denote a discrete phase variable, e.g. $p_t \in \{\text{approach}, \text{align}, \text{grasp}, \dots\}$. The phase is inferred from task progress and contact-related features using a hidden semi-Markov model (HSMM) [40, 46]. The HSMM is trained on demonstration data to capture typical temporal patterns and dwell times of the different phases, and provides at runtime both a most likely phase label p_t and its associated confidence. For each phase $p \in \mathcal{P}$, a quadratic cost

$$J_p = \mathbb{E}\left[\sum_{k=0}^{\infty}\Big((x_k - x_k^{\text{ref}})^\top Q_p (x_k - x_k^{\text{ref}}) + u_k^\top R_p u_k\Big)\right] \tag{4.11}$$

is defined, where x_k^{ref} is a phase-dependent reference (e.g. target pose, desired alignment), and Q_p, R_p are positive semi-definite and positive definite weight matrices, respectively. In early phases such as approach, Q_p penalizes coarse deviations from the target and emphasizes speed, while in alignment and insertion phases, position and orientation accuracy are weighted more strongly and control effort is penalized to encourage smooth, precise motion.

Assuming local linearization of the dynamics (4.5) around μ_t and Gaussian noise, the optimal control law in each phase is given by an LQG controller that combines the EKF with a linear state feedback

$$u_t = -K_p\, \mu_t, \tag{4.12}$$

where K_p is obtained by solving the corresponding algebraic Riccati equation for phase p. In practice, the gain can also be made uncertainty-aware by scaling it as a function of Σ_t, reducing the control aggressiveness when the belief is very uncertain.

In addition to the basic LQG structure, the follower model includes guidance forces that attract the end-effector towards the current task target, as well as resistive forces that discourage unsafe or undesirable motions. The corresponding parameters (e.g. stiffness and damping of the guidance field, scaling factors for assistance and inertia rendering) appear explicitly in the dynamics and cost matrices and are treated as tunable quantities.

4.1.3 Leader model and Bayesian persuasion perspective

The leader problem (4.2) has two distinct levers: (i) dynamic assistance parameters that directly affect the teleoperation dynamics and the follower cost, and (ii) informational parameters that affect the observation model and thus the follower belief. The latter can be interpreted through the lens of Bayesian persuasion. In classical Bayesian persuasion, a principal commits to an information structure—a mapping from states to signals—in order to influence the posterior beliefs and actions of a Bayesian agent in a way that improves the principal's utility. In the present setting, the recommender plays the role of the principal, the human operator the role of the agent, and the EKF provides a lightweight Gaussian approximation of the belief update.

Concretely, let $b_t = (\mu_t, \Sigma_t)$ denote the current belief state and let $\phi_t \in \Phi$ parametrize the information provided at time t through the interface. Examples of ϕ_t include showing or hiding a numerical distance to a target, highlighting an object, or providing an alignment indicator. Given a choice of ϕ_t, the observation model and noise covariance in (4.6) are modified, which in turn changes the EKF update and the evolution of Σ_t. For instance, displaying an exact distance measurement corresponds to injecting an artificial observation with small variance in the corresponding component, reducing the associated uncertainty. If this information is suppressed, the EKF prediction step causes the variance to increase again over time. Because the follower's LQG control law operates on (μ_t, Σ_t), these changes in information directly influence the predicted control actions.

From the leader's perspective, the pair (θ_t, ϕ_t) constitutes an action that jointly specifies dynamic and informational nudges. At each decision time, the recommender evaluates a discrete set of candidate actions $\{(\theta_t^i, \phi_t^i)\}_{i=1}^N$. For each candidate, it simulates the follower dynamics and belief evolution over a finite horizon H using the LQG follower model and the corresponding observation model, obtaining a predicted trajectory $\{x_{t+k}^i, b_{t+k}^i\}_{k=0}^H$ and an associated leader cost

$$J_{\mathrm{l}}^i = \sum_{k=0}^{H} \gamma^k \, \ell_{\mathrm{l}}\Big(x_{t+k}^i, b_{t+k}^i, \theta_t^i, \phi_t^i\Big), \tag{4.13}$$

with discount factor $\gamma \in (0, 1)$ and a stage cost ℓ_{l} that penalizes, for example, task error, high forces, and large belief variance in task-relevant directions, as well as the use of strong assistance or cognitively demanding information. The leader then selects the action with minimal predicted cost,

$$(\theta_t^\star, \phi_t^\star) \in \arg\min_{i \in \{1,\dots,N\}} J_{\mathrm{l}}^i, \tag{4.14}$$

provided that the predicted improvement over a baseline action exceeds a minimum threshold and that budget constraints on assistance and information are respected.

The Gaussian structure of the follower model and the EKF belief update is essential here: it keeps the simulation of belief dynamics tractable and allows the leader to approximate the effect

Figure 4.1: Conceptual overview of the Stackelberg-style recommender. The human follower is modelled as an LQG controller acting on an EKF belief of the teleoperation dynamics. The recommender modifies both dynamic parameters and the information presented to the operator, thereby influencing the belief state and the resulting control actions.

of different informational choices ϕ_t in a lightweight way. While this realization of Bayesian persuasion is still strongly idealized compared to real human cognition, it provides a principled starting point for modeling how targeted information could be used to shape operator beliefs and, through the LQG follower, their control actions in a Stackelberg framework.

A schematic overview of this interaction, indicating where the recommender injects parameter changes and information into the dynamics and the EKF, is provided in Fig. 4.1.

4.2 Practical limitations of the Stackelberg approach

The Stackelberg formulation provides a mathematically clean description of the interaction between recommender and operator, and the implemented LQG follower reproduces qualitative aspects of operator behaviour in teleoperation. Nevertheless, pilot experiments and simulations revealed substantial discrepancies between the predicted effects of nudges and the actual reactions of human operators. These discrepancies ultimately rendered the approach impractical for the present setting.

A first limitation is the modeling error in the human reaction to nudges. The follower model assumes that the operator behaves as an uncertainty-aware LQG controller acting on the EKF belief. While this approximation captures gross features of the motion—such as approaching a target along approximately straight paths and slowing down near contact—it does not accurately reproduce how real operators change their behavior in response to specific changes in information or assistance parameters. In the model, a small modification of the observation noise or a guidance gain leads to a well-defined and smooth change in the optimal control law. In practice, operators often ignore subtle changes, overreact to others, or adapt over time in a way that is not captured by the fixed LQG structure. As a consequence, the predicted benefit of a nudge under the follower model does not reliably reflect its actual effect on human behavior.

A second limitation is a misalignment between what is mathematically beneficial in the LQG sense and what is perceived as helpful by the operator. In the implemented leader cost, reducing belief variance in task-relevant directions is attractive, since it tightens the predicted state distribution and improves the optimal control performance. This leads the recommender to select informational nudges in situations where the belief variance is large, even if the operator is still far away from the target and does not yet rely on precise numerical information. Conversely, in phases where the operator actually cares about fine accuracy, the model variance may already be small, such that the marginal mathematical benefit of additional information is low and the corresponding nudge is not selected. This illustrates a more general issue: the LQG-based objective prioritizes quantities that are convenient to optimize (state deviations and covariances), but these are only imperfect proxies for perceived usefulness, mental workload, or trust.

A third limitation concerns the modeling of different types of nudges and modalities. In the Stackelberg formulation, dynamic and informational nudges enter the model as parameter changes in the dynamics, cost, or observation equations. Distinguishing between modalities (visual, haptic, audio), encoding their different saliencies and cognitive costs, and representing complex cues such as warnings versus continuous indicators requires many additional parameters. These parameters are difficult to identify quantitatively and interact in non-trivial ways with the EKF and LQG components. In practice, this high degree of parameterization made it hard to translate between the abstract mathematical representation of an information structure and the concrete feedback rendered in the interface, leading to nudging policies that were difficult to interpret and tune.

Finally, the follower model implicitly assumes that any information presented to the operator is perceived, correctly interpreted, and integrated into their belief at the time it is displayed. From a modeling perspective this is reasonable, as it preserves the Markov property and allows the EKF to serve as a proxy for the human belief. However, it neglects important aspects of human cognition, such as limited attention, variability in processing speed, and differences between operators in how they interpret specific cues. In reality, feedback that is mathematically optimal according to the model may be overlooked, misunderstood, or arrive at an inopportune moment in the operator's own decision process. These effects introduce an additional, hard-to-model gap between the idealized Bayesian persuasion perspective and the behavior observed in practice.

In combination, these limitations imply that the Stackelberg approach, although fully implemented and computationally feasible at the chosen nudging rate, did not yield robust and consistently helpful recommendations in the teleoperation tasks considered here. The strong dependence on modeling assumptions, the difficulty of aligning mathematical objectives with perceived usefulness, and the gap between idealized information processing and real human cognition motivated a transition to a more data-driven and engineering-oriented design, which is introduced in the following section.

4.3 Contextual bandit-based recommender

Motivated by the practical limitations of the Stackelberg formulation discussed above, the final recommendation system adopts a more pragmatic, data-driven design. Instead of optimizing nudges with respect to a detailed parametric model of the human, it learns and exploits simpler input–output relationships between observable task context and the usefulness of different feedback configurations. This is formalized as a contextual bandit problem, in which the recommender repeatedly observes a context summarizing the current situation and selects nudges with the aim of maximizing task-related reward.

4.3.1 Overview and problem formulation

At a high level, the recommendation system is decomposed into four components:

1. an *intent inference* module that estimates which object or sub-task the operator is currently targeting using object detections and end-effector motion;

2. a *sequence planner* that computes an efficient assembly plan and identifies admissible and optimal next parts;
3. a *task coordinator* that combines intent and plan information, tracks task progress, and determines when and how to intervene;
4. a *recommender* that selects dynamic and informational nudges based on a contextual bandit formulation.

The first three modules provide structured, task-level information that is independent of the specific learning algorithm and could in principle be reused by different recommenders. The contextual bandit layer then maps this structured context to concrete parameter and interface changes.

Nudging decisions are made at discrete decision times $t \in \{0, \Delta t_{\text{nudge}}, 2\Delta t_{\text{nudge}}, \dots\}$, which are slow compared to the underlying teleoperation and control loops. At each such time, the system constructs a context vector

$$x_t \in \mathcal{X}, \tag{4.15}$$

that encodes relevant information about

- the *task state* (current phase, which parts have already been placed, admissible next parts, distance to targets);
- the *robot and interaction state* (end-effector pose and velocity, contact forces and torques, workspace margins);
- the *operator intent* (most likely target object, associated probability, entropy of the intent distribution);
- the *interface state* (currently active nudges, remaining budgets and cool-downs for different modalities).

Based on this context, the recommender chooses a pair of actions

$$a_t^{\text{dyn}} \in \mathcal{A}^{\text{dyn}}, \qquad a_t^{\text{info}} \in \mathcal{A}^{\text{info}}, \tag{4.16}$$

where $\mathcal{A}^{\text{dyn}}$ is a finite set of dynamic parameter configurations (e.g. combinations of guidance strength, damping levels, workspace scaling), and $\mathcal{A}^{\text{info}}$ is a finite set of informational cues (e.g. object highlights, distance indicators, warnings, alignment hints). The dynamic action a_t^{dyn} is realized by updating the corresponding parameters of the haptic and robot controllers, while the informational action a_t^{info} is realized by updating the head-up display and other interface elements.

The effect of these actions on task performance is not known a priori and depends on the operator. Instead of modeling this dependence explicitly, the system treats it as an unknown reward function. For the dynamic channel, a scalar reward

$$r_t^{\text{dyn}} = r^{\text{dyn}}(x_t, a_t^{\text{dyn}}) \tag{4.17}$$

is defined over short episodes following a decision, combining measures such as progress towards the target, smoothness of motion, and effort (e.g. integrated handle force). For the informational channel, a reward

$$r_t^{\text{info}} = r^{\text{info}}(x_t, a_t^{\text{info}}) \tag{4.18}$$

reflects task-level effects of cues and warnings, such as avoiding inadmissible actions, reducing unnecessary detours, or preventing high contact forces.

This leads to two separate contextual bandit problems. For the dynamic assistance,

$$\max_{\pi^{\text{dyn}}} \mathbb{E}\left[\sum_t r_t^{\text{dyn}}\right] \quad \text{s.t.} \quad a_t^{\text{dyn}} = \pi^{\text{dyn}}(x_t), \tag{4.19}$$

and for the informational nudges,

$$\max_{\pi^{\text{info}}} \mathbb{E}\left[\sum_t r_t^{\text{info}}\right] \quad \text{s.t.} \quad a_t^{\text{info}} = \pi^{\text{info}}(x_t), \tag{4.20}$$

where π^{dyn} and π^{info} denote policies that map contexts to actions. The expectation is taken over the stochastic evolution of the human–robot system and, in learning mode, over any exploration in the policies themselves.

In this work, both problems are solved using linear contextual bandits with Thompson sampling. For each action, a linear model of the expected reward as a function of the context is maintained, together with a Bayesian posterior over its parameters. At decision time, a parameter sample is drawn from the posterior, the corresponding expected reward is evaluated for all admissible actions, and the action with highest sampled value is selected. This balances exploitation of actions that are known to work well with exploration of less well-understood settings, while remaining computationally lightweight. Importantly, the dynamic and informational bandits are trained and operated separately, reflecting their different action spaces and reward structures, but they share the same underlying context x_t provided by the intent inference, planner, and coordinator.

4.3.2 Intent inference, sequence planning, and task coordination

The contextual bandit does not operate directly on raw sensor data, but on a structured context that already encodes hypotheses about operator intent and task structure. This structure is provided by three components: an intent inference module, a sequence planner, and a task coordinator that combines both and exposes a consistent target state to the recommender.

Intent inference

The intent inference module estimates which object the operator is currently targeting based on object detections and end-effector motion. Let $\mathcal{O} = \{1, \ldots, K\}$ denote the set of candidate parts in the workspace, and let $o_t \in \mathcal{O}$ be the (unobserved) index of the part the operator intends to pick at time t. The inference maintains a discrete belief

$$\pi_t \in \Delta^{K-1}, \qquad \pi_t(k) = \mathbb{P}(o_t = k \mid \text{observations up to time } t),$$

where Δ^{K-1} denotes the $(K-1)$-simplex.

At each time step, the module receives the current end-effector position x_t^{ee}, velocity v_t^{ee}, and the estimated positions $x_t^{(k)}$ of all detected parts $k \in \mathcal{O}$ from the AprilTag-based perception system.

For each candidate k, an observation likelihood $L_t(k)$ is computed from geometric features such as the distance $\|x_t^{\text{ee}} - x_t^{(k)}\|$ and the alignment between the motion direction v_t^{ee} and the vector pointing towards the part. Parts that are close to the end-effector and approximately in the current motion direction obtain higher likelihood values.

Temporal consistency is encoded through a transition matrix $T \in \mathbb{R}^{K \times K}$ with entries

$$T_{ij} = \mathbb{P}(o_{t+1} = j \mid o_t = i),$$

constructed either as a uniform switching model or as a sequential model that favours staying with the current part and moving to the next one in a predefined order. The belief is then updated by a standard Bayesian filter

$$\tilde{\pi}_{t+1}(j) = L_{t+1}(j) \sum_{i=1}^{K} T_{ij} \pi_t(i), \qquad \pi_{t+1} = \frac{\tilde{\pi}_{t+1}}{\sum_{k=1}^{K} \tilde{\pi}_{t+1}(k)}. \tag{4.21}$$

The most likely target part at time t is given by

$$\hat{o}_t = \arg\max_{k \in \mathcal{O}} \pi_t(k),$$

and its associated confidence is $\pi_t(\hat{o}_t)$. Uncertainty about the intent is quantified by the Shannon entropy

$$H(\pi_t) = -\sum_{k=1}^{K} \pi_t(k) \log \pi_t(k).$$

Both $\hat{o}_t$ and $H(\pi_t)$ are exposed to the higher-level modules and to the contextual bandit. In particular, low entropy and high confidence indicate that the operator is strongly committed to a specific part, whereas high entropy indicates that they are still undecided or switching between alternatives.

Sequence planner

The sequence planner provides a normative reference for an efficient assembly sequence. For the gearbox assembly task, precedence constraints between parts are given (e.g. certain shafts must be placed before their covers), and each part $k \in \mathcal{O}$ has a known pick pose p_k^{pick} and place pose p_k^{place} in a common world frame. Given the current end-effector pose x^{ee} and the set of already placed parts $S \subseteq \mathcal{O}$, the planner considers as admissible next parts the set

$$\mathcal{A}_{\text{adm}}(S) = \{k \in \mathcal{O} \setminus S \mid \text{all predecessors of } k \text{ are in } S\}.$$

To approximate path efficiency, a step cost is defined for picking and placing a single part k from a current pose x,

$$c(x, k) = \|x - p_k^{\text{pick}}\| + \|p_k^{\text{pick}} - p_k^{\text{place}}\|, \tag{4.22}$$

where $\|\cdot\|$ denotes the Euclidean distance in Cartesian space. This cost approximates the path length of moving from the current pose to the pick position and then to the place position.

Given an initial pose x_0 and an initial set of placed parts S_0, the planner computes a sequence $\sigma^\star = (k_1, \ldots, k_m)$ of the remaining parts that minimizes the cumulative cost

$$J_{\text{seq}}(x_0, S_0, \sigma) = \sum_{j=1}^{m} c(x_{j-1}, k_j), \tag{4.23}$$

subject to $k_j \in \mathcal{A}_{\text{adm}}(S_{j-1})$ and the update rules

$$S_j = S_{j-1} \cup \{k_j\}, \qquad x_j = p_{k_j}^{\text{place}}.$$

This optimization is solved by dynamic programming with memorization over the discrete state (S, x), reusing previously computed subproblems. The resulting sequence $\sigma^\star$ provides an approximate shortest path assembly plan that respects the precedence constraints.

In addition to producing a current recommended next part $k^\star \in \mathcal{A}_{\text{adm}}(S)$, the planner exposes several query functions:

- *Admissibility:* whether a candidate part k is admissible given S.
- *Sequence cost:* the total cost $J_{\text{seq}}(x_0, S_0, \sigma)$ for an arbitrary candidate sequence σ.
- *Marginal cost:* the cost if a particular part is chosen first, $c(x_0, k)$ + future cost.

The latter is used to quantify the difference between the currently optimal sequence and an alternative sequence consistent with the operator's intent, enabling explicit reasoning about how suboptimal a preferred human choice would be in terms of path length.

Task coordination

The task coordinator links the intent inference and the sequence planner and provides a consistent target description and warnings to the recommender and user interface. It maintains the current grasp state (e.g. pick vs. place), the set of placed parts, and the active target pose for the robot.

At each update, the coordinator receives:

- the belief π_t and its entropy $H(\pi_t)$ from the intent inference;
- the set of admissible next parts $\mathcal{A}_{\text{adm}}(S_t)$, the planner's recommended next part $k_t^\star$, and sequence cost information from the sequence planner;
- the current object poses and the end-effector pose from the perception and teleoperation system.

It then performs the following steps:

1. *Target selection.* A preliminary target part is chosen based on the inferred intent $\hat{o}_t$ and the planner's recommendation $k_t^\star$. To avoid rapid switching, a dwell time mechanism is used: the active target is only updated if a new candidate remains dominant for a minimum time interval. The target's position and orientation are taken from the corresponding object poses.
2. *Admissibility and warning generation.* If the intent belief is confident (low entropy and high $\pi_t(\hat{o}_t)$) and the inferred target $\hat{o}_t$ is not admissible, the coordinator flags a *hard warning* indicating that the desired part cannot be placed at this time (e.g. due to missing

predecessors). If $\hat{o}_t$ is admissible but differs from $k_t^\star$, the coordinator queries the sequence planner for the cost difference between following the optimal plan and following the operator's preferred choice. If this difference exceeds a threshold, a *soft suggestion* can be issued to highlight the planner's recommended part and explain that it would lead to a shorter overall path.

3. *State update.* When the system detects that a part has been successfully placed (e.g. via dedicated state flags or perception), the coordinator updates the placed set S_t and triggers a re-planning step in the sequence planner. For insertion tasks, a similar mechanism is used to update the next insertion target based on the current state.

The outputs of the coordinator are (i) the current target pose and identifier, (ii) an admissibility flag for the inferred intent, (iii) optional warning or suggestion objects including the part name and pose, and (iv) measures of plan sub-optimality (e.g. cost difference between plans). These signals are integrated into the contextual feature vector x_t and provide the basis for higher-level informational nudges (such as highlighting inadmissible parts or recommending more efficient alternatives) and for the dynamic assistance bandit (through the target pose and phase information). This design separates task-level reasoning from the learning algorithm and ensures that the bandit operates on semantically meaningful context features rather than raw sensor measurements.

4.3.3 Linear contextual bandit with Thompson sampling

The design goal for the recommender is to adapt feedback to the operator and task without relying on a brittle parametric model of human behavior. At the same time, the system should remain simple enough to be tuned and analyzed, and it should not require large amounts of training data. A full reinforcement learning formulation with arbitrary state dynamics would be unnecessarily complex in this setting: nudging decisions are made at a relatively slow rate, the effect of each nudge is primarily local in time, and a rich, hand-engineered context x_t is available that already captures most of the relevant history (Section 4.3.2).

Contextual bandits provide a natural compromise [3, 31]. In a contextual bandit, the learner repeatedly observes a context x_t, chooses an action a_t, and observes a scalar reward r_t that depends on (x_t, a_t) but not on the full future state trajectory. There is no explicit modeling of long-term state evolution beyond what is encoded in the context itself. This matches the structure of the present problem: for each decision, the recommender sees the current task phase, target, intent confidence, and interaction state, selects a dynamic or informational nudge, and later observes whether this choice improved local performance.

In this work, both the dynamic assistance and the informational recommender are implemented as *linear contextual bandits* with Thompson sampling [4, 25, 32, 47]. For each bandit, it is assumed that the expected reward of an action a in context x_t can be approximated by a linear model

$$\mathbb{E}\Big[r_t \mid x_t, a_t = a\Big] \approx x_t^\top \theta_a, \tag{4.24}$$

where $\theta_a \in \mathbb{R}^d$ is an unknown parameter vector associated with action a and d is the dimension of the feature vector x_t. The randomness in the reward is captured by an additive noise term,

$$r_t = x_t^\top \theta_{a_t} + \varepsilon_t, \tag{4.25}$$

with ε_t modeled as zero-mean noise.

Bayesian linear model and posterior updates

The bandit maintains, for each action $a \in \mathcal{A}$, a Bayesian linear regression model. A Gaussian prior

$$\theta_a \sim \mathcal{N}(\mu_{a,0}, \Sigma_{a,0}) \tag{4.26}$$

is placed on the parameter vector, with prior mean $\mu_{a,0}$ and covariance $\Sigma_{a,0}$. Given a sequence of observations $\{(x_\tau, r_\tau)\}$ for which $a_\tau = a$, the posterior over θ_a remains Gaussian and can be updated in closed form. For numerical convenience, the implementation maintains, for each action a, the sufficient statistics

$$A_a = \lambda I_d + \sum_{\tau:a_\tau=a} x_\tau x_\tau^\top, \qquad b_a = \sum_{\tau:a_\tau=a} r_\tau x_\tau, \tag{4.27}$$

where $\lambda > 0$ is a regularisation parameter and I_d is the $d \times d$ identity matrix. The posterior mean and covariance are then

$$\mu_a = A_a^{-1} b_a, \qquad \Sigma_a = A_a^{-1}. \tag{4.28}$$

These expressions correspond to a ridge-regularised least-squares estimate with an implicit Gaussian prior and are computationally lightweight: each update only requires rank-one corrections of A_a and b_a.

Action selection by Thompson sampling

At a decision time t with context x_t, the bandit uses Thompson sampling to choose an action. For each action $a \in \mathcal{A}$, a parameter sample

$$\tilde{\theta}_a \sim \mathcal{N}(\mu_a, \Sigma_a) \tag{4.29}$$

is drawn from the current posterior. The sampled parameters define a sampled action value

$$q_a(x_t) = x_t^\top \tilde{\theta}_a, \tag{4.30}$$

and the selected action is

$$a_t \in \arg\max_{a \in \mathcal{A}} q_a(x_t). \tag{4.31}$$

This procedure balances exploitation and exploration in a principled way: actions whose parameters are well known have concentrated posteriors, leading to small variability in $q_a(x_t)$ and thus near-greedy behavior, whereas actions with high uncertainty have broader posteriors, which occasionally produce large sampled values and are therefore explored.

In the present system, separate instances of this linear Thompson sampling scheme are maintained for different task phases and for the two nudging channels (dynamic assistance and informational cues). For example, the effect of changing workspace scaling during an “approach” phase is different from that during an “align” phase, and the action sets are not identical. Maintaining distinct bandit models per phase allows the system to capture these differences without artificially enlarging the context vector.

Initialization from a deterministic policy

A purely online contextual bandit would typically start from an uninformative prior, such as $\mu_{a,0} = 0$ and $\Sigma_{a,0} = \lambda^{-1} I_d$, and gradually learn the reward model from experience. In the teleoperation setting considered here, this approach is problematic for two reasons. First, naive exploration can lead to clearly undesirable behavior (e.g. excessively high resistance or distracting information) and thus to a poor user experience. Second, the amount of interaction time available for learning in a user study is limited, so that a bandit starting from scratch would not converge to a meaningful policy.

To address this, the bandit models are initialized from offline data collected under a hand-crafted deterministic policy. This policy encodes heuristic knowledge about which feedback configurations are reasonable in which contexts, for example increasing guidance strength close to the target or activating haptic resistance when high velocities are detected near obstacles. Running this policy on the teleoperation system and recording the resulting contexts x_τ, chosen actions a_τ, and observed rewards r_τ yields a dataset

$$\mathcal{D} = \{(x_\tau, a_\tau, r_\tau)\}_{\tau=1}^{N}.$$

The sufficient statistics (A_a, b_a) for each action a are then initialized as

$$A_a = \lambda I_d + \sum_{\tau : a_\tau = a} x_\tau x_\tau^\top, \qquad b_a = \sum_{\tau : a_\tau = a} r_\tau x_\tau, \tag{4.32}$$

which correspond to the posterior of the Bayesian linear model after observing $\mathcal{D}$. In practice, these matrices are precomputed and stored, and the online bandit loads them as its initial state.

This initialization has two advantages. First, it ensures that the bandit starts from a policy that is already sensible according to the deterministic baseline, avoiding obviously poor actions early on. Second, it allows the bandit to focus its exploration on local refinements of this baseline, adapting to individual operators and task variations by updating the posterior as new rewards are observed. In this sense, the deterministic policy serves as an expert-like prior, and the bandit performs active learning around it rather than learning from a uniform prior.

Rationale for the chosen structure

The resulting structure—two separate linear contextual bandits, each with phase-specific models and initialized from deterministic policies—reflects the different nature of dynamic and informational nudges, and the constraints of the teleoperation scenario. Dynamic assistance acts continuously on the haptic and robot dynamics and can be evaluated using smoothness- and effort-based rewards defined over short episodes. Informational cues are discrete, sparse events that are constrained by cognitive load and interface design, and their effect is closely tied to task structure and intent. Modeling both channels within a single monolithic learner would require a heterogeneous action space and heterogeneous reward definitions and proved brittle in preliminary experiments. By contrast, separate bandits with shared context but tailored action sets and rewards provided a good balance between expressiveness, interpretability, and robustness in the implemented system.

4.3.4 Dynamic assistance bandit

The dynamic assistance bandit adapts continuous interaction parameters of the teleoperation interface. Its goal is to support the operator by improving efficiency and smoothness of motion, while avoiding excessive effort and preserving the overall feeling of control. The available actions modify three scalar parameters:

- s_g : scaling of the guidance field that attracts the end-effector towards the current target;
- s_{ws} : workspace scaling factor between handle motion and robot motion;
- s_{hap} : scaling of resistive haptic forces, used to damp unsafe motions.

These parameters appear in the teleoperation dynamics and follower model as defined in Section 4.1.2, and are exposed to the recommender through the operator parameter vector.

Features

At each decision time, the dynamic bandit receives a reduced set of continuous features extracted from the current state and recent history. For a given target position x^{target} and end-effector position x^{ee}, the following features are computed:

- normalized distance to target,

$$\text{dist}_n = \min\left(1,\ \frac{\|x^{\text{ee}} - x^{\text{target}}\|}{d_r^{\max}}\right),$$

  where $d_r^{\max}$ is a saturation distance;
- progress towards the target over time, expressed as a filtered, normalized scalar prog_n;
- progress oscillation, approximated by the time derivative of progress prog_dot_n and filtered to capture back-and-forth motions;
- reachability margin, measuring how close the handle is to the physical limits of its workspace when mapped to the current workspace scaling s_{ws};
- handle room, defined from the normalized handle position relative to the center of the device workspace;
- normalized effort,

$$\text{effort}_n = \min\left(1,\ \frac{\|f^{\text{hap}}\|}{f^{\max}}\right),$$

  where f^{hap} is the force at the handle and $f^{\max}$ is a saturation value.

These raw quantities are low-pass filtered online with exponential filters to avoid overly reactive behavior and to capture trends over short episodes. The feature vector x_t for the bandit is then formed by stacking

$$x_t = \left[\text{dist}_n,\ \text{prog}_n,\ \text{prog\_dot}_n,\ \text{reachability},\ \text{room\_handle},\ \text{effort}_n\right]^\top,$$

ordered according to a configuration file.

Action space and phase dependence

The dynamic assistance bandit maintains different action sets for different task phases (e.g. "approach", "align", "insert"). In each phase, the action space $\mathcal{A}_p^{\text{dyn}}$ consists of discrete configurations of the three parameters $(s_g, s_{\text{ws}}, s_{\text{hap}})$ around the current operating point. For example, during an approach phase, the actions may mainly modify workspace scaling and haptic resistance to keep the handle within a comfortable range, whereas during alignment the actions focus on guidance strength and local damping.

To ensure predictable behavior, actions are grouped and subject to cool-downs: if a parameter group (e.g. guidance strength) has been modified recently, the same group cannot be changed again for a minimum time interval. This avoids rapid switching that would be confusing for the operator. The bandit thus effectively chooses between a small number of phase-dependent parameter updates that respect these constraints.

Reward design

The reward for the dynamic channel is defined over short episodes and combines local progress towards the target, oscillations, and effort. At each tick, given the current features, an instantaneous reward

$$r_t^{\text{dyn}} = w_p \, \text{prog}_n - w_o \, |\text{prog\_dot}_n| - w_e \, \text{effort}_n \tag{4.33}$$

is computed, with positive weights w_p, w_o, w_e chosen empirically and the result clipped to a fixed range. This expresses the intuitive trade-off that the bandit should encourage steady progress (high prog_n), penalize back-and-forth movements (large $|\text{prog\_dot}_n|$), and avoid high sustained effort at the handle.

In the implementation, actions are evaluated over episodes of fixed duration $\Delta t_{\text{episode}}$ by integrating or averaging r_t^{dyn} over the episode. When an episode ends (for example, because a parameter group leaves cool-down or a phase changes), the cumulative reward is attributed to the action that initiated the episode, and the corresponding linear bandit parameters are updated as described in Section 4.3.3. This yields a compact closed loop: the bandit uses features x_t to select $(s_g, s_{\text{ws}}, s_{\text{hap}})$, observes the resulting local motion pattern, and updates its belief about which parameter settings are beneficial in which contexts.

4.3.5 Informational bandit

The informational bandit governs discrete cues presented to the operator via the head-up display or, in principle, audio feedback. Its goal is to provide timely, concise information that helps the operator make better decisions (e.g. choosing an admissible and efficient next part, avoiding excessive forces), without overloading their attention.

Whereas the dynamic assistance acts continuously on the physical interaction, informational nudges are sparse events with explicit modality and cognitive cost. The informational bandit therefore operates under budget and cool-down constraints that limit how much visual or audio information can be presented over time.

Features

The informational bandit uses a richer context, combining interaction, intent, and planning information. Typical feature components include:

- normalized end-effector speed and contact force/torque;
- distance and orientation error relative to the current target (e.g. for alignment in pre-grasp and insertion phases);
- handle workspace margin, indicating how close the handle is to its limits;
- predicted target part $\hat{o}_t$, its confidence $\pi_t(\hat{o}_t)$, and the entropy $H(\pi_t)$ of the intent belief;
- planner recommendation $k_t^\star$ and an indicator whether the inferred target $\hat{o}_t$ is admissible;
- measures of plan sub-optimality, such as the estimated cost difference between following $k_t^\star$ and following the operator's preferred part;
- remaining visual and audio "budgets" $C_{\text{rem}}^{\text{vis}}$, $C_{\text{rem}}^{\text{aud}}$ and per-modality cool-downs.

These quantities are normalized using configuration-dependent scales (e.g. maximum speed, maximum admissible force) and are combined into a feature vector x_t that is shared with the informational bandit model.

Action space and cost budgets

Informational actions are defined as symbols corresponding to specific cues in the interface, such as

- `target_highlight`: highlight the recommended part in the scene or on the HUD;
- `distance`: display a numerical distance or progress bar towards a part or insertion target;
- `alignment_linear` / `alignment_rotational`: show alignment indicators during pre-grasp or insertion;
- `speed_warning`, `force_warning`, `torque_warning`: warn about excessive speed or loads;
- `admissible_warning`: indicate that the currently intended part is not admissible given the assembly state;
- `twin_target_locked` or similar task-specific messages indicating that the avatar is in a constrained configuration;
- `clutch`: suggest using the clutch when the handle is close to workspace limits.

Each informational action is associated with:

- one or more modalities (visual, audio);
- a cognitive cost per modality (how demanding the cue is);
- a phase-dependent maximum cost $C_{\max}$ and a per-modality budget and cool-down.

At decision time, the bandit only considers actions whose modalities are currently allowed (no active cool-down) and whose cost does not exceed the remaining budget. This enforces sparsity and prevents the system from flooding the operator with cues.

Scoring and selection

Given the current context x_t, an internal value or score is computed for each candidate informational action. Conceptually, this score reflects how beneficial the action is expected to be in the current situation, balanced against its cost. In the fully learning-based formulation, this expectation is provided by the linear contextual bandit model from Section 4.3.3, i.e. the bandit estimates

$$\mathbb{E}\left[r_t^{\text{info}} \mid x_t, a_t^{\text{info}} = a\right] \approx x_t^\top \theta_a$$

from data and uses Thompson sampling to select an action.

In practice, this learning is combined with task-specific structure. For example, during approach and alignment phases, highlighting the planner-recommended target and warning about inadmissible intended parts receives high priority when the intent is confident and entropy is low. During pre-grasp, alignment cues and force/torque warnings become more important, and distance indicators are de-emphasized if the operator is already close to the target. The implementation encodes such priorities in per-action weightings and combines them with the bandit value estimates and the cost/budget constraints to compute a score for each candidate. If the best score exceeds a small threshold, the corresponding informational action is issued; otherwise, no new information is presented.

Reward design

The informational reward r_t^{info} is less directly tied to instantaneous motion than the dynamic reward and is instead defined around discrete events and task outcomes. Typical contributions include:

- positive reward when a warning prevents an inadmissible action (e.g. trying to place a part out of order);
- positive reward when following a suggestion leads to a sequence closer to the planner-optimal one (reduced detour cost);
- small positive reward for providing alignment or distance cues that precede successful grasps or insertions;
- negative reward for presenting cues during periods of high workload that do not correlate with measurable improvements (proxy for unnecessary distraction).

In the implementation used for the experiments, these rewards are approximated using proxy measures derived from the logged task execution (e.g. whether inadmissible attempts occurred, how often warnings coincided with corrections, and simple path-length measures). For the purpose of this thesis, the key point is that the informational bandit operates under explicit constraints on bandwidth and cognitive load, and that its learning or prioritization focuses on when and which cues are truly beneficial given the operator's intent and the task plan.

Overall, the dual-bandit architecture—with a continuous, trajectory-based reward for dynamic assistance and an event-based, task-level reward for informational cues—proved more robust and interpretable than a single monolithic learner. It allows the system to adapt physical assistance and information presentation in a coordinated but structurally separated way, while reusing the same intent and planning context.

4.4 Integration with the teleoperation architecture

The contextual bandit-based recommender is implemented as part of a central coordination process that interfaces with the existing teleoperation architecture. This architecture comprises separate processes for the haptic device and robot control, the VR-based visual interface, and perception (object detection), each with its own local state machine. The recommender does not modify these low-level state machines; instead, it operates on a slower time scale, reads their state, and writes back high-level parameter updates and informational cues.

4.4.1 Software architecture and state machines

On the teleoperation side, the haptic device, the robot controller, and the VR interface each maintain an internal state machine with states such as *idle*, *calibration*, *manual teleoperation*, and *task execution*. A separate Python-based coordination node (the "operator assistant") maintains a higher-level state machine that encodes the experiment flow, including task-specific states such as *gearbox assembly*, *insertion task*, and *evaluation*.

The recommender is active only in task-execution states for which it has been configured, and only when the underlying devices report ready and healthy operating conditions. In other states, all bandit outputs are ignored and the system falls back to a fixed "base" parameter set corresponding to standard teleoperation without nudging. This separation ensures that safety-critical aspects such as emergency stops, joint limits, and device faults remain under the control of the low-level state machines and are not affected by the recommender.

Within a task state, the coordination node periodically:

- collects the current device states (handle pose and force, end-effector pose, twist, and contact wrench, interface mode, task progress flags);
- calls the intent inference and sequence planner to update the current target, admissibility, and plan information;
- calls the task coordinator to update the placed-part set, generate warnings or suggestions, and provide a consistent target pose;
- constructs the context vector x_t for the dynamic and informational bandits;
- queries the dynamic and informational bandits for actions a_t^{dyn} and a_t^{info};
- writes back updated operator parameters (e.g. s_g, s_{ws}, s_{hap}) and informational cues into the shared data structures that are consumed by the haptic and VR processes.

All communication between processes is realised via UDP messages on a local wired network. For latency-critical signals (handle pose and end-effector pose), separate high-frequency channels are used; the recommender operates only on aggregated state and does not interfere with these low-level loops.

4.4.2 Timing and data flow

The teleoperation control loops run at high frequency: the haptic rendering and robot impedance controller operate in the range of several hundred hertz to one kilohertz, and the VR rendering loop runs at video frame rate. In contrast, the recommender operates at a nudging rate of approximately

$$f_{\text{nudge}} = \frac{1}{\Delta t_{\text{nudge}}} \in [1, 5]\,\text{Hz},$$

depending on configuration. This separation of time scales ensures that changes in assistance and information occur slowly enough to be perceived and interpreted by the operator, and that the computational cost of the bandit and planning logic remains negligible compared to the real-time control loops.

At each nudging step, the coordinator integrates sensor and state information over the preceding interval to compute the features required by the bandits. For the dynamic assistance bandit, this includes filtered progress, oscillations, and effort over the episode since the last parameter change. For the informational bandit, it includes event counts (e.g. inadmissible attempts, high-force events) and updated plan sub-optimality measures. The corresponding rewards are computed and attributed to the actions that were active during the episode, and the bandit statistics (A_a, b_a) are updated accordingly. In the experimental configuration used for the user study, the bandits themselves were initialized from offline data and operated predominantly in exploitation mode, with limited or disabled exploration to avoid confusing the participants.

4.4.3 Safety and fallback behavior

All parameters that the recommender may change are constrained to lie within pre-defined safe intervals. For example, workspace scaling s_{ws} is bounded to ensure that the handle is never mapped outside the robot workspace, guidance strength s_g is limited to avoid excessive attraction forces, and resistive scaling s_{hap} is restricted to remain within comfortable force levels. These bounds are enforced both in the bandit configuration and in the low-level controllers.

If the coordination node detects missing or stale data (e.g. loss of perception, invalid intent estimate, communication timeouts), or if the experiment state leaves the supported task phases, the recommender is temporarily disabled. The system then falls back to the base teleoperation mode with fixed assistance and no additional informational cues, until normal operation is restored. This design ensures that the recommender is strictly additive: it can improve guidance when active, but its failure or deactivation does not compromise the basic teleoperation functionality.

Finally, all nudging decisions and relevant context variables are logged at the nudging rate in synchrony with the sensor data and video streams. These logs are used offline to compute the rewards for the bandits, to analyze the behavior of the recommender, and to correlate its actions with the operators' performance and subjective ratings in the user study.

4.5 Summary

This chapter presented the design of the recommendation system that provides nudging-style shared control in the teleoperation setup. The initial Stackelberg formulation modeled the operator as a phase-dependent LQG follower acting on an EKF belief, and cast the recommender as a leader that optimists dynamic and informational parameters in a bi-level optimization framework. While mathematically appealing, this approach suffered from significant modeling errors and misalignment between mathematically beneficial quantities and perceived usefulness.

To address these limitations, the final recommender adopts a contextual bandit formulation with two separate bandits for dynamic assistance and informational cues. Both bandits operate on a structured context that includes intent inference, task planning, and interaction features, and they are initialized from deterministic policies to provide sensible behavior from the start. The resulting dual-bandit architecture is computationally lightweight, interpretable, and compatible with the constraints of the teleoperation system. It integrates into the existing software architecture via a central coordination node and modifies only high-level parameters and interface elements, preserving safety and operator authority. The next chapter describes the experimental study used to evaluate the effect of the different feedback modes on task performance and operator experience.

5 Experimental Methodology

This chapter describes the experimental methodology used to evaluate the proposed teleoperation architecture and integrated recommender system. The main objective is to investigate how different feedback modes influence task performance, safety, and operator experience in representative manipulation scenarios.

Each experimental session starts with a short training task that allows operators to familiarise themselves with the avatar system, the user interface, and the available assistance functions. The actual evaluation is then carried out on two teleoperated manipulation tasks: a structured *gearbox assembly* task that emphasizes sequencing and efficiency, and a *tactile insertion* task that relies strongly on haptic feedback and precise alignment. For each task, human operators perform repeated trials under multiple feedback configurations, ranging from a minimal baseline to conditions with dynamic assistance and informational nudging. During all trials we record objective performance and safety metrics, as well as subjective ratings of workload, perceived control, and perceived usefulness of the assistance.

The remainder of this chapter is structured as follows. Section 5.1 introduces the training and evaluation tasks and their physical setup. Section 5.2 defines the feedback modes under comparison. Section 5.3 describes the participants and experimental procedure. The measurement signals and evaluation metrics are detailed in Section 5.4, and Section 5.5 explains the data processing and analysis methods.

5.1 Tasks and scenarios

We evaluate the system using one short training task and two evaluation tasks that capture both structured sequencing and contact-rich interaction.

5.1.1 Training task

Before the main trials, each participant performs a simple pick-and-place task in the robot workspace. Coloured cubes are placed on the table, and the participant is instructed to pick up the cubes and place them on a black rectangular area within the workspace. No specific ordering, time limit, or performance target is given. Instead, the purpose of this task is for the participant to familiarize themselves with the teleoperation setup and to explore all available control and assistance functions.

5.1.2 Gearbox assembly

The first evaluation task is a toy gearbox assembly using 3D-printed components. The scene comprises a rigidly mounted base plate (fixture) and five movable parts: two shafts of different diameters and three gears of different sizes. At the beginning of each trial, the movable parts are placed at accessible but otherwise arbitrary locations within the robot workspace. All movable parts are instrumented with AprilTag markers, enabling the perception module to estimate their poses online and providing state information for the recommender. Figure 5.1 illustrates the task by showing the initial part layout and the target assembled configuration.

To facilitate grasping and reliable detection, the two shafts are initially placed upright in 3D-printed holders. If a shaft is dropped during a trial, it is manually returned to its holder. Similarly, if a gear or shaft falls out of the reachable workspace or ends up in a configuration that is difficult to grasp, it is manually repositioned to a reachable and graspable pose. This intervention avoids coupling task performance to workspace boundaries or recovery from unrecoverable configurations and keeps the focus on the assembly process itself.

The gearbox can only be assembled in configurations that respect certain dependencies between parts (e.g., gears must be placed on the appropriate shafts and into the base plate in a compatible order), but there exist multiple valid assembly sequences. During the initial briefing, participants are informed about the intended final configuration and a reasonable assembly order so that the evaluation concentrates on sequential decision-making and execution rather than on discovering feasible assembly strategies. The mechanical tolerances of the printed parts are relatively large, such that the insertions themselves are not technically demanding. Instead, the main challenge of this task lies in selecting an admissible sequence of actions, coordinating the placement of parts, and maintaining generally efficient behavior.

Overall, the gearbox assembly task provides a structured, multi-step scenario that is representative of teleoperated industrial assembly. It allows us to study how different feedback modes influence sequential decision-making, the avoidance of inadmissible intermediate states, and the efficiency of the operator's interaction with the environment.

5.1.3 Tactile insertion

The second evaluation task is a peg-in-hole insertion that emphasizes precise alignment and contact-rich interaction. The setup consists of a rigidly mounted receptacle and three cylindrical pegs of different diameters selected from a larger set. The clearances between the pegs and the matching hole are small enough that successful insertion requires accurate positioning and, in particular, precise rotational alignment. Even when performed directly by hand, the insertion is not straightforward and typically requires careful tactile exploration. An overview of the peg-in-hole insertion setup is shown in Fig. 5.2.

In contrast to the gearbox assembly task, the picking phase is omitted here: at the beginning of each insertion trial, a peg is already grasped in the avatar gripper. The operator's task is to insert the peg into the hole using teleoperation. After completing an insertion attempt, the operator presses a dedicated home button, which returns the robot arm and the haptic device to a standardized start configuration and resets the relevant control parameters. This procedure ensures consistent initial conditions across insertions and feedback modes. If the

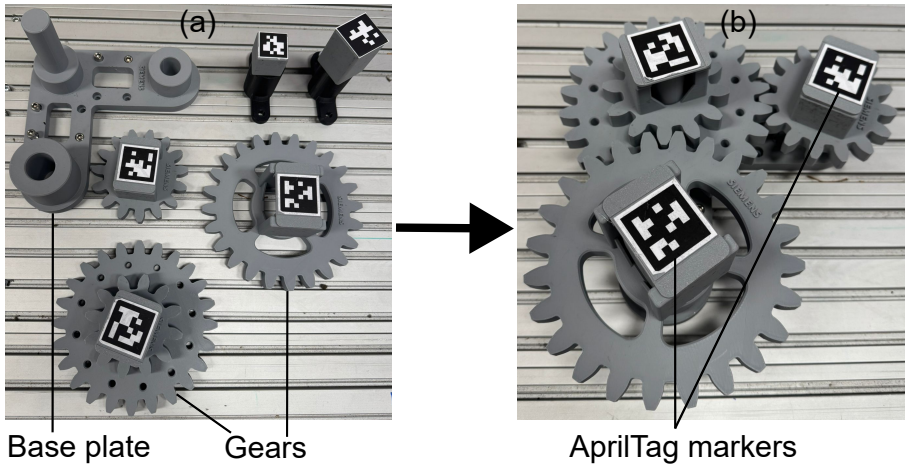


Figure 5.1: Gearbox assembly task visualization. (a) Initial layout of the fixture (base plate) and movable components (gears and shafts) within the workspace. (b) Target assembled configuration. AprilTag markers attached to the movable parts enable online pose estimation used by the perception and recommender modules.

peg is accidentally dropped or the gripper is opened before a proper insertion is achieved, the experimenter places the peg back into the gripper to restore a comparable starting state.

By varying the peg diameter, the task spans different levels of insertion difficulty while keeping the overall structure simple: the goal and the required part are always unambiguous, and the main challenge lies in managing contact forces, exploiting haptic feedback, and achieving the necessary alignment. As such, the tactile insertion task is well suited to study how different feedback modes influence force interaction quality, stability, and the effectiveness of haptic cues in teleoperated manipulation.

5.2 Feedback modes

To investigate the effect of different levels and types of assistance, we define six feedback modes that systematically vary the available information channels and the degree of automatic dynamic support. All modes provide the same teleoperation interface and avatar system; only the feedback and assistance configuration is changed.

Manual In the *manual* mode, the operator receives only visual feedback from the head-mounted display. No haptic rendering, audio cues, or supplementary information are provided. The operator performs the task without any system assistance, which makes this mode comparable to a basic teleoperation baseline.

Base The *base* mode augments visual feedback with audio signals and contact force rendering. Haptic feedback reflects interaction forces at the end-effector, but task-related parameters such as workspace scaling and haptic resistance are fixed. This configuration approximates human operation with access to natural sensor cues (vision, sound, and touch) but without higher-level guidance.

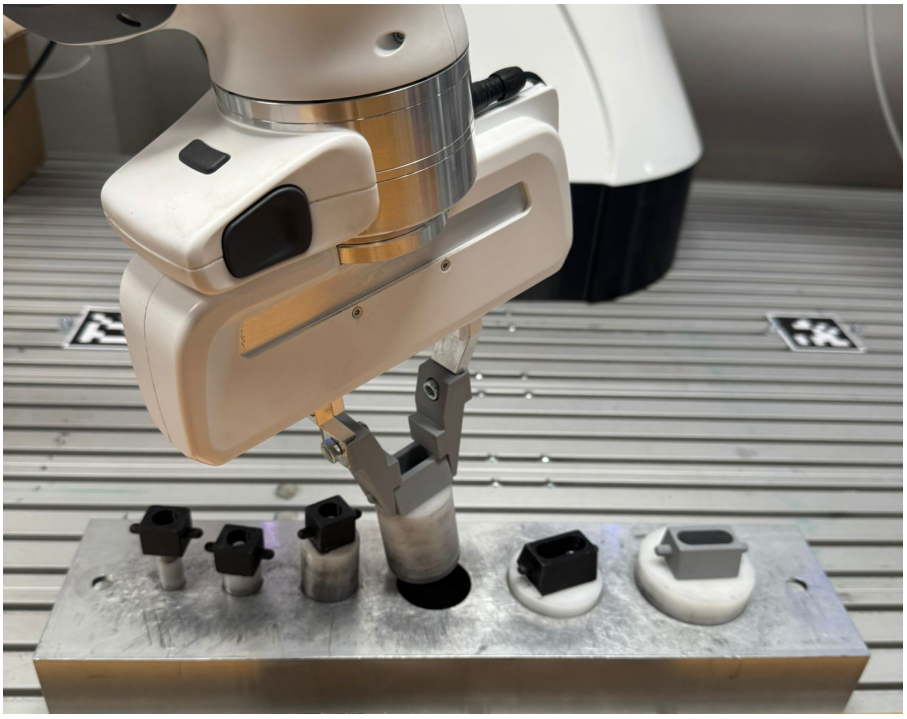


Figure 5.2: The robot aligns a cylindrical peg held in a parallel gripper with the fixture opening and performs a contact-rich peg-in-hole insertion under the different assistance modes.

Enhanced base The *enhanced base* mode includes all feedback of the base mode and adds standard task-relevant information, such as height, force, or torque indicators in the user interface. In addition, the operator can manually adjust selected teleoperation parameters (e.g., workspace scaling or controller gains). This mode provides richer situational awareness and explicit control over the assistance settings, but no automated adaptation or recommendations.

Pure informational assistance The *pure info* mode builds on the enhanced base configuration and activates the informational channel of the recommender system. Higher-level cues are presented that are predictive or contextual with respect to the current task stage, for example guidance during approach, indications of insertion readiness, or alignment hints. Dynamic parameters are not adapted automatically. This mode is used to assess whether additional task-relevant information alone can improve operator performance and behaviour.

Pure dynamic assistance The *pure dynamics* mode focuses on automatic dynamic support. Starting from the enhanced base configuration, the system adapts teleoperation parameters such as workspace scaling and haptic resistance based on the current task context and interaction forces, and provides haptic insertion assistance to improve alignment. Informational cues from the recommender are not shown explicitly in the user interface. This mode corresponds to a form of shared control, where assistance is primarily conveyed through changes in the underlying dynamics rather than through explicit instructions.

Full assistance The *full* mode combines the informational and dynamic channels. It includes the higher-level informational cues of the pure info mode and the automatic parameter adaptation and haptic insertion support of the pure dynamics mode. This configuration represents the maximum level of assistance considered in this work and is used to study whether the simultaneous provision of explicit guidance and implicit dynamic support leads to further benefits or introduces additional cognitive load for the operator.

5.3 Participants and procedure

Ten participants took part in the study. All were novel users of the teleoperation system, with ages ranging from 24 to 29 years. Each experimental session lasted approximately two hours, including the introductory briefing, the training task, both evaluation tasks, and short breaks. Before the start of the experiment, participants were informed about the purpose and structure of the study and signed a written informed consent form.

At the beginning of each session, participants received a slide-based briefing. This briefing covered safe behavior in the laboratory, the teleoperation interface and avatar system, the head-up display and user interface elements, the experimental tasks, and the NASA-TLX workload questionnaire used after the trials [18, 37]. Participants were informed which signals would be recorded, that the data would be used for analysis, and that all data would be stored under an anonymous operator identifier rather than by name.

After the briefing, participants performed the training task described in Section 5.1.1. During this phase, they were explicitly encouraged to explore all available control and assistance functions and to familiarize themselves with the system. The experimenter provided verbal guidance to ensure that the functionality of the interface and feedback modes was understood. For the subsequent experimental trials on the gearbox assembly and tactile insertion tasks, the experimenter did not interact with the participants, except for providing initial task instructions and handling technical issues, in order to keep the conditions comparable across subjects.

The main experiment comprised the gearbox assembly task (Section 5.1.2) and the tactile insertion task (Section 5.1.3), each executed under the six feedback modes defined in Section 5.2. The order of the two tasks (gearbox first vs. insertion first) was randomized across participants. For the gearbox assembly, each feedback mode was repeated twice, resulting in 12 assembly trials per participant. For the tactile insertion, each feedback mode was combined with three pegs of different diameters, yielding 18 insertion trials per participant. Within each task, the sequence of feedback modes was randomized under the constraint that the same mode was not presented in consecutive trials.

Each gearbox trial had a time limit of 5 min and each insertion trial a time limit of 3 min, with a visual 30 s warning in both cases. Trials in which the time limit was exceeded were considered unsuccessful. If objects were dropped or moved into unrecoverable configurations, they were repositioned as described in Sections 5.1.2 and 5.1.3 to avoid confounding task performance with workspace boundary effects. After every trial, participants completed a NASA-TLX questionnaire and were instructed to rate the workload solely with respect to the most recent trial.

5.4 Measurements

During all trials we recorded both low-level sensor signals and high-level experiment annotations for subsequent offline analysis. The robot arm, the avatar head, and the haptic device continuously logged their sensor data to timestamped text files at 200 Hz. Each file contains a header followed by semicolon-separated numeric values, which can be read directly with standard CSV tools (e.g. `pandas`). The logged signals include joint states, end-effector poses and wrenches, gripper state, and haptic device positions and forces, as well as relevant teleoperation mode flags.

In addition, the recommender system recorded its internal view of the task in a separate log stream, also at 200 Hz. These logs are stored as JSONL files and contain the incoming sensor and perception streams, detected object poses, inferred targets and task sequences, issued informational cues and dynamic parameter adaptations, operator inputs processed by the recommender, and trial-level annotations such as task start and end times and success flags. Together, the low-level and recommender logs provide a detailed, time-synchronized record of both the physical interaction and the assistance behavior.

To capture the experiments from multiple visual perspectives, we recorded three video streams: a workspace view from the perception camera at 15 Hz, the operator's view through the VR headset including overlaid user interface elements, audio signals, and subtitles for the text-to-speech output, and an external camera monitoring the operator's body movements during the tasks. These recordings are used for qualitative inspection and to resolve ambiguities in the sensor logs.

Subjective workload was assessed after every trial using the NASA-TLX questionnaire. Note that the TLX performance subscale reflects perceived performance (lower scores indicate better perceived performance) and should not be confused with objective task metrics. The responses are stored in a CSV file containing a running trial index, the feedback mode identifier, the individual item ratings, and the anonymous operator identifier. All measurement files (sensor logs, recommender logs, video recordings, and questionnaires) are stored under the respective operator ID, enabling consistent multi modal post-processing while preserving anonymity. The derivation of quantitative performance and safety metrics from these measurements is described in Section 5.5.

5.5 Data processing and analysis

All recorded data were processed offline using custom Python scripts based on `pandas` and `numpy`. The robot arm, avatar head, haptic device, and recommender logs share a common timestamp originating from the recommender input stream, so that all numeric signals can be interpreted on a single time axis. For each experimental session, the semicolon-separated sensor log files and the JSONL recommender logs were parsed into a unified time-series representation at 200 Hz. Trial boundaries (start, end, task, and feedback mode) were obtained from explicit annotations in the recommender log. Video recordings were used for qualitative inspection and to resolve ambiguities in the sensor data but were not analyzed quantitatively.

From the synchronized time series we derived performance, interaction, and workload measures

on a per-trial basis. Basic performance indicators include task completion time and a binary success label, determined from the trial start and end markers and the task-specific time limits. Kinematic and interaction descriptors are obtained from the end-effector trajectory and wrench, such as path length, velocity and acceleration statistics, and summaries of contact forces and torques. For the tactile insertion task we additionally distinguish between free-space motion and contact phases, based on simple force thresholds, and compute phase-specific statistics where required.

Subjective workload is evaluated after every trial using the NASA-TLX questionnaire. The responses are stored in a CSV file together with the trial index, feedback mode, and anonymous operator identifier. From these data we compute an overall TLX score as well as selected sub-scale scores for each trial.

All metrics are first computed per trial. For the gearbox task, they are then aggregated per participant and feedback mode over the two repetitions; for the insertion task, they are aggregated per participant and feedback mode over the three pegs. Group-level comparisons between feedback modes are based on these per-participant averages and are primarily reported using descriptive statistics and visualizations. Where appropriate, we also explore simple relationships between objective performance, interaction measures, and subjective workload to better understand how the different feedback modes influence operator behavior.

6 Results

This chapter presents the experimental results for the proposed teleoperation architecture and integrated recommender system. We first report task-specific outcomes for the gearbox assembly and tactile insertion scenarios, focusing on task performance, interaction characteristics, and subjective workload across the six feedback modes. We then analyze cross-cutting patterns, comparing workload between tasks and examining how efficiency and interaction quality relate to subjective experience.

Section 6.1 summarizes the results for the gearbox assembly task. Section 6.2 reports the corresponding results for the tactile insertion task, including a phase-specific analysis of free-space motion and contact behavior. Section 6.3 provides cross-condition analyses that relate completion time, contact features, and workload, and compares the effect of the feedback modes across both tasks.

6.1 Gearbox assembly

The gearbox assembly task required participants to pick and place a set of parts into a housing under time pressure, following a specific assembly structure. The task combines reaching and placing actions, intermittent contacts, and moderate precision, and therefore probes both path-efficiency and interaction dynamics.

6.1.1 Task performance

Figure 6.1 summarizes task time and success rate across the six feedback modes.[1]

Manual operation (visual feedback only) shows the lowest success rate and the largest median completion time, with many trials approaching the time limit. The Baseline mode (haptics, audio, and basic interface) improves performance moderately, but still exhibits substantial variability between participants. All three recommender-related modes (Enhanced baseline, Pure information, Pure dynamic, and especially Full) shift the distribution towards shorter task times and higher success rates. In particular, the information- and dynamics-enhanced modes reduce the fraction of timeouts and cluster more tightly around lower median times, indicating that the additional guidance helps participants follow a more efficient assembly strategy.

[1] Unless stated otherwise, all plots in this chapter aggregate over participants and repetitions; error bars show the standard error of the mean.

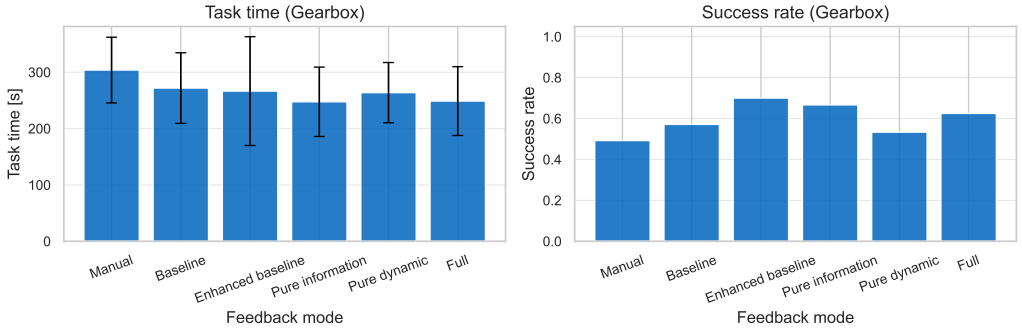
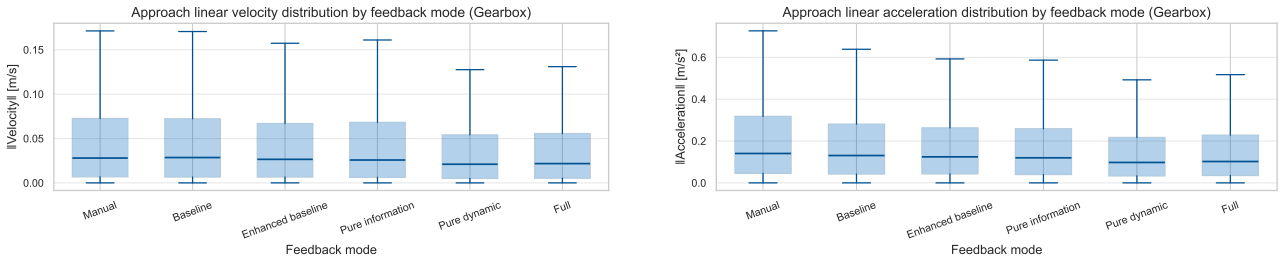


Figure 6.1: Task time and success rate for the gearbox assembly task across feedback modes.



(a) Approach linear velocity.

(b) Approach linear acceleration.

Figure 6.2: Kinematic features for the approach phase in the gearbox task by feedback mode.

6.1.2 Interaction characteristics

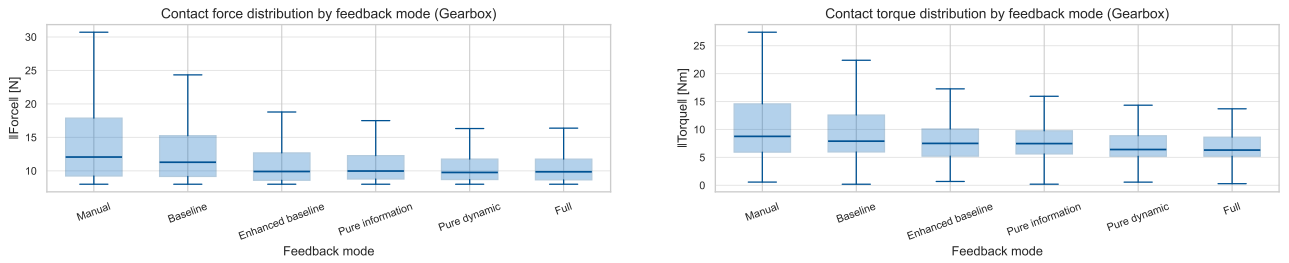
To analyze how the different feedback modes affect movement and contact behavior, we extracted kinematic and force features from the gearbox trials. The approach phase covers the motion from leaving the previous placement to arriving in the vicinity of the next part or placement location. The contact phase aggregates periods in which the robot end-effector is in contact with objects or fixtures.

Figure 6.2 shows the distributions of approach velocity and acceleration across modes.

Across modes, the median approach velocities remain similar, indicating that the overall movement tempo is comparable. However, Manual and Baseline exhibit broader distributions and more pronounced upper tails in acceleration, reflecting more abrupt corrections and less regular motion. The recommender-driven modes (Enhanced baseline, Pure information, Pure dynamic, Full) tend to reduce the highest acceleration peaks while keeping median values close to the unassisted modes. This suggests that participants move with a similar average speed but require fewer sudden adjustments when additional guidance is available.

Contact behavior is summarized in Fig. 6.3, which shows the distributions of contact forces and torques.

Manual operation again shows the largest upper tails in both force and torque, indicating more frequent high-load contacts during placement. Baseline reduces these peaks slightly, but the three enhanced modes show a clearer shift towards lower 95th-percentile forces and torques and narrower interquartile ranges. Dynamic assistance (Pure dynamic and Full) appears particularly effective in reducing the strongest contact events, consistent with the idea that guidance fields and



(a) Contact force. (b) Contact torque.

Figure 6.3: Contact force and torque distributions for the gearbox task by feedback mode.

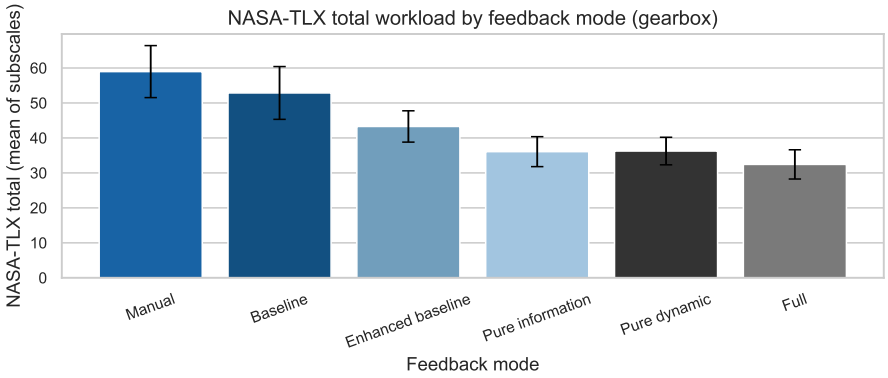


Figure 6.4: NASA–TLX overall workload for the gearbox task by feedback mode (mean over participants, error bars: standard error).

damping adjustments smooth the interaction and discourage forceful corrections. Informational cues alone already help to avoid extreme contacts, but the combination of information and dynamics in the Full mode yields the most consistently gentle interaction.

6.1.3 Subjective workload

Figure 6.4 shows the NASA–TLX overall workload score for the gearbox task, computed as the mean of the six subscales.

Manual control yields the highest workload, with Baseline slightly lower but still clearly above the enhanced modes. Enhanced baseline, Pure information, Pure dynamic, and Full all reduce the average TLX total by roughly 10–20 points on the 0–100 scale compared to Manual. The three recommender-related modes lie in a similar low-workload range, with Pure information and Pure dynamic achieving the lowest mean scores. A within-subject z-normalization of the TLX total (Fig. 6.5) confirms the same ordering, indicating that this pattern is consistent across participants despite individual differences in absolute ratings.

The subscale analysis in Fig. 6.6 shows that this reduction is driven mainly by mental demand, effort, and frustration.

Mental demand and frustration are highest in the Manual condition (approximately 60–65 on average) and decrease monotonically as additional guidance and information are provided, reaching values in the low 30s for the enhanced modes. Effort follows a similar pattern. Physical

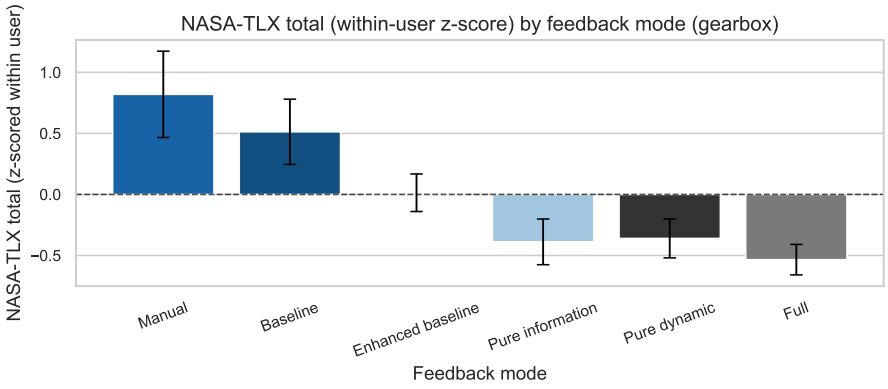


Figure 6.5: NASA–TLX overall workload for the gearbox task, z-scored within participants.

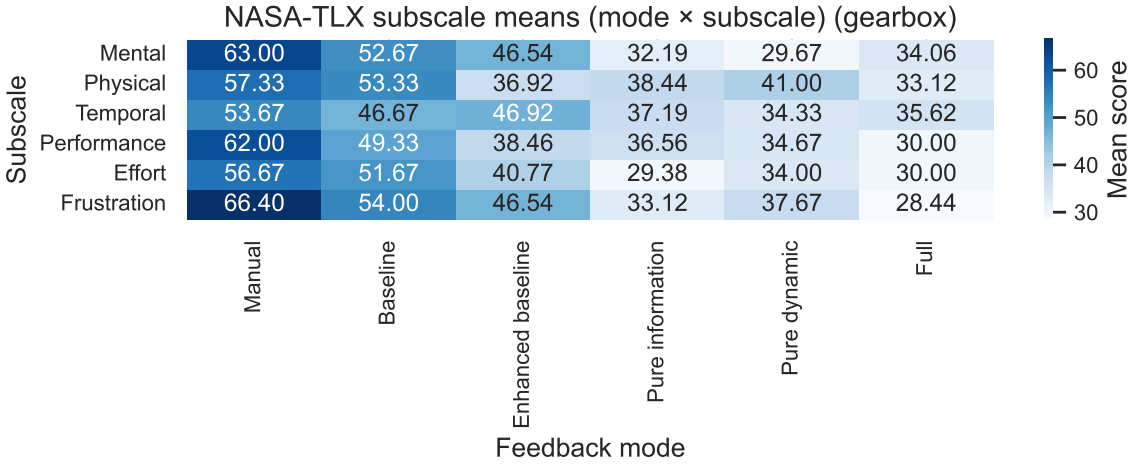


Figure 6.6: NASA–TLX subscale means for the gearbox task (mode × subscale).

and temporal demand also decrease, although the relative reduction is smaller than for the cognitive dimensions.

The comparatively modest change in temporal demand may be attributed to the task design and presentation of time constraints. Time pressure becomes salient primarily toward the end of a trial; as long as a sufficient time margin is perceived, participants may largely disregard remaining time. Moreover, temporal feedback was intentionally unobtrusive, consisting of a late warning shortly before timeout and a peripheral time bar rather than a centrally prominent cue. Repeated exposure to the task across trials may have further reduced sensitivity to time constraints, particularly in the absence of strong incentives linked to completion time. As a result, perceived temporal demand appears less responsive to the assistance modes than the cognitive workload dimensions.

A paired per-user comparison of TLX total scores (Fig. 6.7) further indicates that most participants consistently rate the enhanced modes lower than the Manual and Baseline conditions.

Overall, the gearbox results support the hypothesis that targeted feedback and assistance can improve performance while also reducing perceived workload. Informational and dynamic modes yield similar benefits in this task, with no clear indication that combining them in the Full mode leads to additional subjective cost.

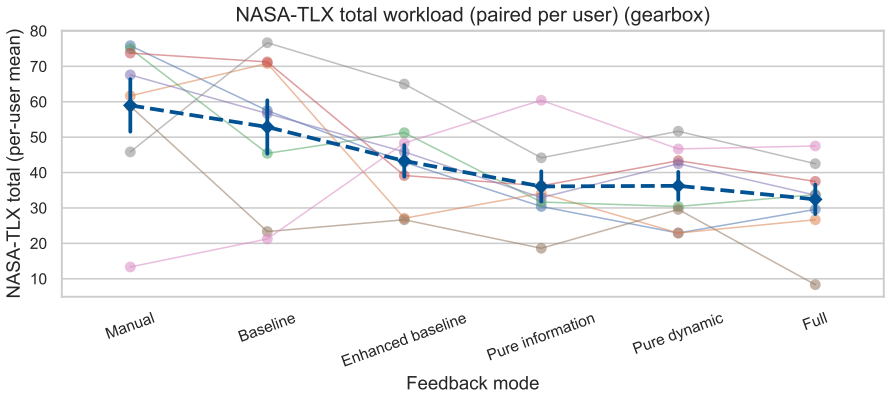


Figure 6.7: NASA–TLX overall workload for the gearbox task, paired per participant across feedback modes.

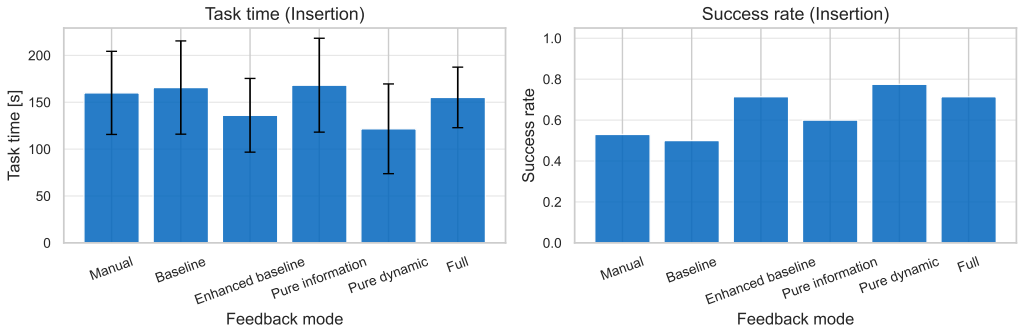


Figure 6.8: Task time and success rate for the tactile insertion task across feedback modes.

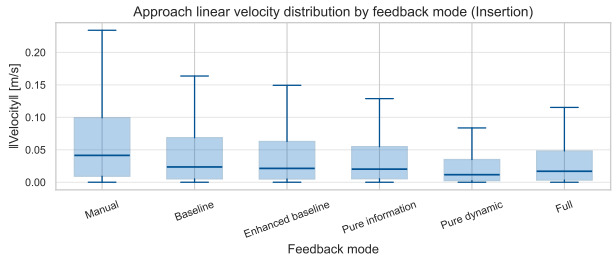
6.2 Tactile insertion

The tactile insertion task required participants to insert three cylindrical pegs into tight-clearance holes within a fixed time limit. Compared to the gearbox assembly, this task is more contact-rich and accuracy-dominated [55], and therefore places stronger emphasis on fine haptic feedback and precise guidance during contact.

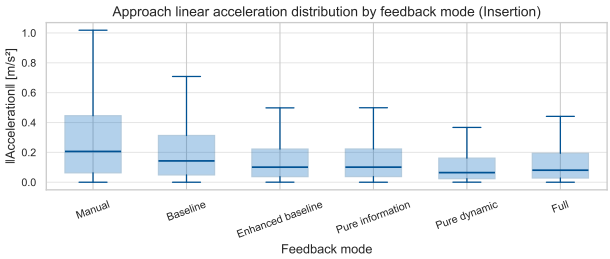
6.2.1 Task performance

Figure 6.8 summarizes task time and success rate across the six feedback modes.

As in the gearbox task, Manual and Baseline show the lowest success rates and the longest median task times, with many trials ending in timeouts. Adding enhanced interface information or dynamic assistance increases the fraction of successful insertions and reduces completion times. The three recommender-related modes cluster clearly on the “faster and more successful” side compared to Manual and Baseline, although differences between Enhanced baseline, Pure information, Pure dynamic, and Full are smaller than the gap to the two baselines. This again supports the first hypothesis: targeted feedback improves objective performance relative to a purely manual visual-only setup.

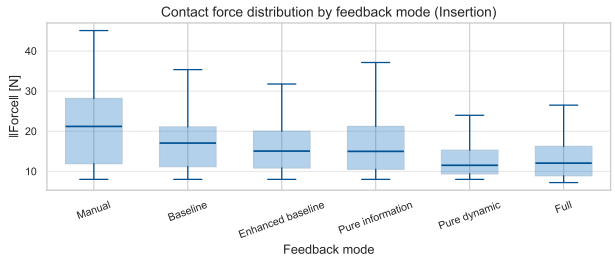


(a) Approach linear velocity.

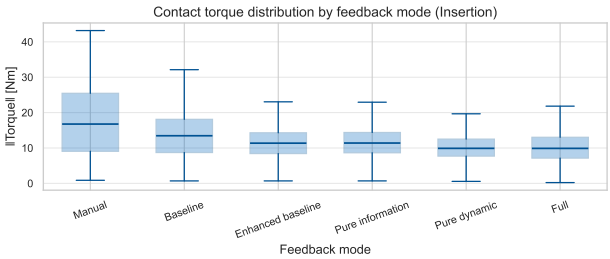


(b) Approach linear acceleration.

Figure 6.9: Kinematic features for the insertion task by feedback mode.



(a) Contact force.



(b) Contact torque.

Figure 6.10: Contact force and torque distributions for the insertion task by feedback mode.

6.2.2 Phase-specific kinematics and forces

For the insertion task, we perform a phase-specific analysis of movement and contact. The insertion trajectory is segmented into approach (free-space motion towards the fixture) and contact (in-contact exploration and alignment). For each phase, robust statistics of linear velocity, linear acceleration, contact forces, and torques are computed.

Figure 6.9 shows the distributions of approach velocity and acceleration.

Median approach velocities are similar across feedback modes, indicating that participants maintain a comparable nominal speed during free-space motion toward the fixture. However, Manual and Baseline conditions exhibit broader distributions and higher upper outliers, whereas the enhanced modes consistently reduce peak velocities and concentrate the distributions. This pattern suggests that additional guidance primarily limits excessive approach speeds rather than altering typical motion behavior. By attenuating high-velocity outliers, the assisted modes promote more controlled and regular approaches, reducing the likelihood of abrupt corrections immediately before contact.

Figure 6.10 shows contact forces and torques during the contact phase.

Here the trend is even more pronounced than in the gearbox task. Manual and Baseline show the largest upper tails in both force and torque, indicating frequent high-load contacts and stronger misalignment during insertion. All enhanced modes shift the distributions downward and narrow the interquartile ranges. The dynamic modes (Pure dynamic and Full) yield the lowest 95th-percentile forces and torques, consistent with the intended effect of the haptic insertion helper and guidance fields [2, 6, 24, 45]. Informational modes already reduce some of the strongest contact events, but the combination of information and dynamics appears

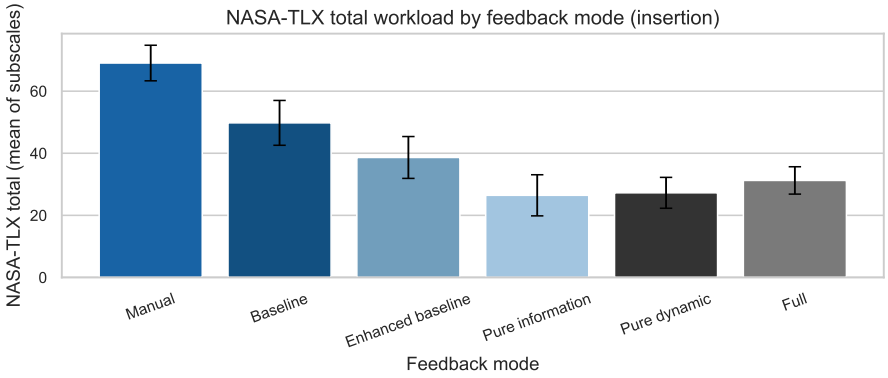


Figure 6.11: NASA–TLX overall workload for the insertion task by feedback mode (mean over participants, error bars: standard error).

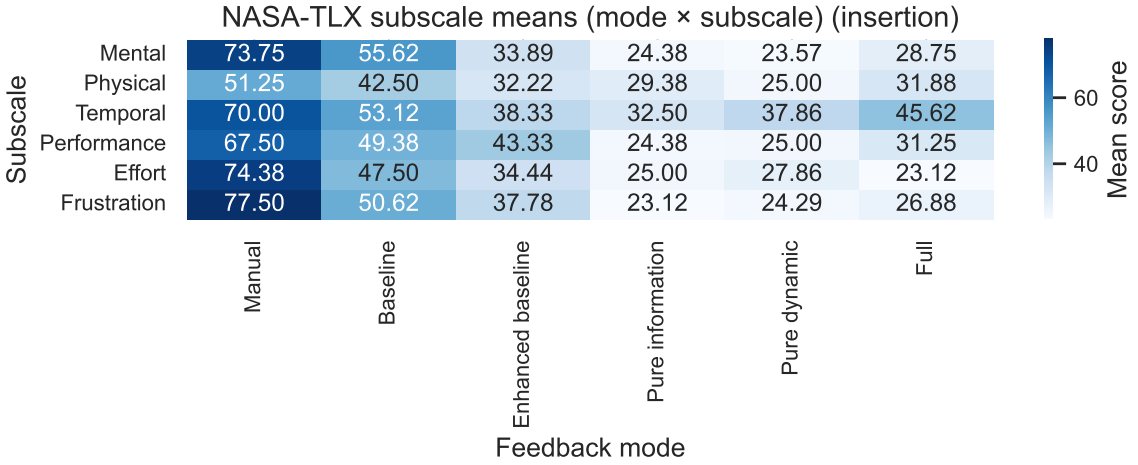


Figure 6.12: NASA–TLX subscale means for the insertion task (mode × subscale).

particularly effective in this high-precision, contact-rich task.

6.2.3 Subjective workload

Figure 6.11 shows the NASA–TLX overall workload for the insertion task.

Manual control again produces the highest workload, with Baseline somewhat lower but still clearly above the enhanced modes. Enhanced baseline, Pure information, Pure dynamic, and Full all reduce the TLX total by roughly 20–40 points relative to Manual. Pure information and Pure dynamic achieve the lowest mean scores, with Full in a similar low-workload range. Within-subject z-normalization confirms that this ordering is robust across participants.

The subscale analysis in Fig. 6.12 reveals that, for insertion, mental demand, effort, and frustration are particularly strongly affected by the feedback mode.

In the Manual condition, mental demand and frustration are both in the 70–80 range on average and decrease monotonically as information and assistance are added, reaching values around 25–30 for the enhanced modes. Effort shows a similar reduction. Physical and temporal demand

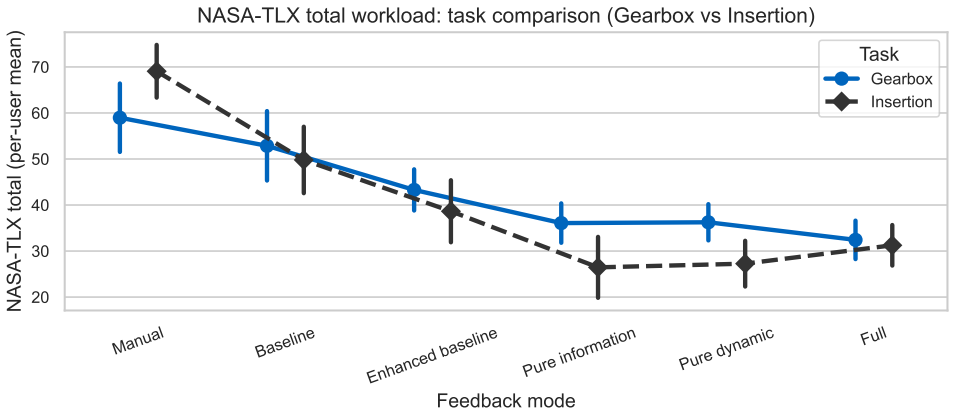


Figure 6.13: NASA–TLX overall workload for gearbox vs. insertion across feedback modes (mean ± standard error).

also drop, but less dramatically. Compared to the gearbox task, the relative benefit of the enhanced modes on mental demand and frustration is even larger here, matching the intuition that precise contact interactions are especially cognitively taxing when performed without guidance.

Overall, the insertion results indicate that the proposed feedback modes do not only improve success rates and reduce contact forces and torques; they also make the high-precision insertion task feel substantially less demanding and frustrating. Informational and dynamic nudges both contribute to this effect, with no evidence that combining them in the Full mode introduces an additional workload penalty in this task.

6.3 Cross-condition analyses

The previous sections considered the gearbox and insertion tasks separately. This section compares patterns across tasks and examines how objective measures of efficiency and interaction quality relate to subjective workload.

6.3.1 Task-level workload comparison

Figure 6.13 compares the NASA–TLX overall workload between the gearbox and insertion tasks across feedback modes, while Fig. 6.14 shows the corresponding subscales.

Across all modes, the insertion task is perceived as more demanding than the gearbox assembly, reflecting its higher precision requirements and sustained contact. Nevertheless, the *relative* pattern across modes is remarkably consistent between tasks: Manual and Baseline yield the highest workload, while Enhanced baseline, Pure information, Pure dynamic, and Full all reduce the TLX total. The three recommender-driven modes cluster together in the low-workload region for both tasks, indicating that the benefits of informational and dynamic nudging generalize from a mixed reaching/placement task to a more contact-critical insertion scenario.

The subscale comparison shows that the main differences between tasks lie in mental demand,

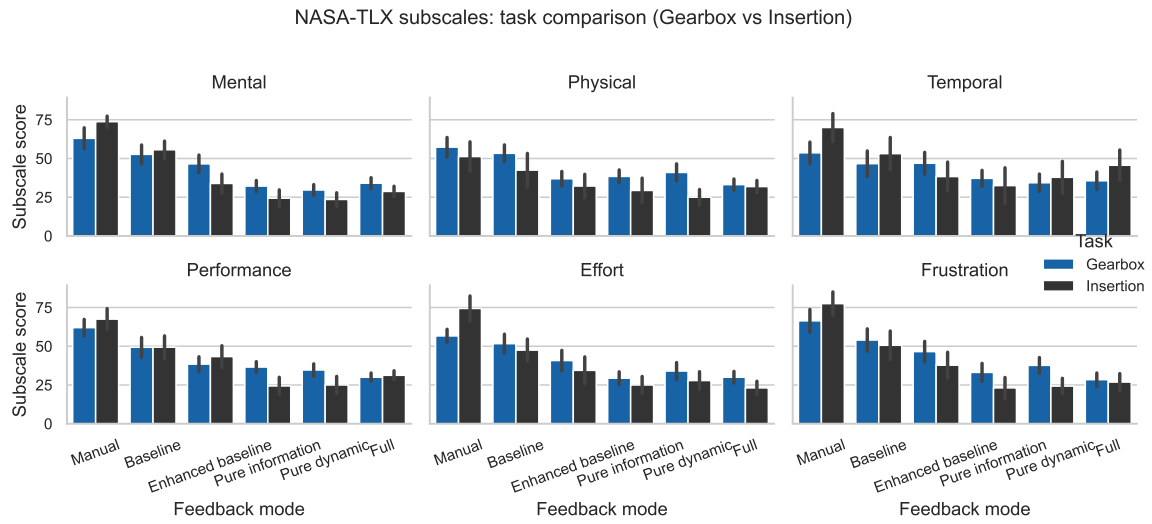
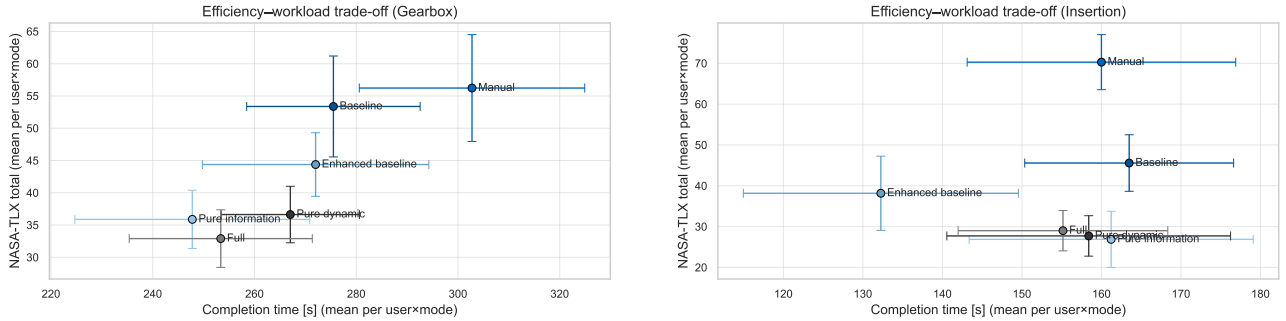


Figure 6.14: NASA–TLX subscale comparison between gearbox and insertion tasks (mean ± standard error).



(a) Gearbox

(b) Insertion

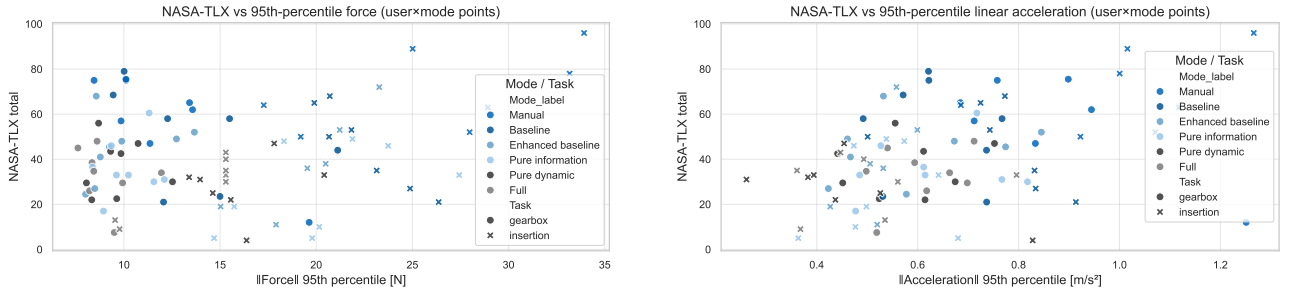
Figure 6.15: Efficiency–workload trade-offs across modes. Each point shows mode-wise means of completion time and TLX total (per participant×mode averages).

effort, and frustration, which are all higher for insertion. For both tasks, however, these cognitive dimensions decrease monotonically from Manual through Baseline to the enhanced modes. Physical and temporal demand exhibit similar but smaller trends. This supports the interpretation that the recommender primarily helps by reducing cognitive and affective load, rather than by simply changing the physical difficulty of the tasks.

6.3.2 Efficiency–workload trade-offs

To examine how efficiency and workload trade off across modes, we computed, for each participant and mode, the mean completion time and mean TLX total per task. The mode-wise averages are summarized in the scatter plots in Fig. 6.15.

In the gearbox task, Manual and Baseline occupy the upper-right region (longer times and higher workload), whereas the enhanced modes mainly move downward in TLX, with only modest reductions in time. Mode-wise means illustrate this pattern: Manual averages $302.8 \pm 58.6\,\mathrm{s}$



(a) TLX vs. 95th-percentile force

(b) TLX vs. 95th-percentile acceleration

Figure 6.16: Relationship between subjective workload and interaction features (user×mode points, colored by mode and task).

and a TLX total of 56.2 ± 21.9, while Full averages 253.4 ± 50.9 s and 32.9 ± 12.6. Enhanced baseline, Pure information, and Pure dynamic lie in between but all clearly below Manual in TLX. A per-user Pearson correlation of $r = 0.57$ between TLX and time indicates that, for gearbox, participants who take longer also tend to report higher workload.

For the insertion task, the enhanced modes again move primarily downward in TLX rather than strongly left in time. Manual averages 160.9 ± 44.7 s and a TLX total of 70.3 ± 17.8, whereas Pure information and Pure dynamic achieve substantially lower workload (26.9 ± 19.4 and 27.7 ± 13.1) with comparable average times (161.2 ± 50.5 s and 158.4 ± 47.2 s). The correlation between TLX and time across participant×mode points is weak ($r = 0.12$), suggesting that in this precision-limited task perceived difficulty is not primarily driven by how long the trial takes.

Taken together, these results suggest that the proposed feedback modes primarily improve the *experience* of teleoperation—lower workload and more comfortable interaction—while leaving completion times similar or modestly improved. There is no evidence of a strong trade-off where reduced workload would systematically come at the cost of slower operation.

6.3.3 Relationships between interaction quality and workload

Finally, we examined how subjective workload relates to interaction-quality features derived from robot sensors. For each participant and mode, we computed the 95th-percentile of the contact force magnitude and linear acceleration, and compared these to the mean TLX total.

Figure 6.16 shows the relationship between TLX total and the 95th-percentile force and acceleration across all participant×mode points for both tasks.

For the gearbox task, TLX shows no clear linear relationship with the 95th-percentile contact force (Pearson $r = -0.03$) and only a weak positive correlation with acceleration ($r = 0.12$). This is consistent with the more intermittent and moderate contacts in this task: participants may not be strongly aware of small differences in peak forces as long as the assembly remains successful.

In the insertion task, by contrast, workload correlates strongly with both contact force and acceleration: TLX vs. force yields $r = 0.62$, and TLX vs. acceleration $r = 0.55$. Participants who experienced higher peak forces and more jerky motion during insertion also reported

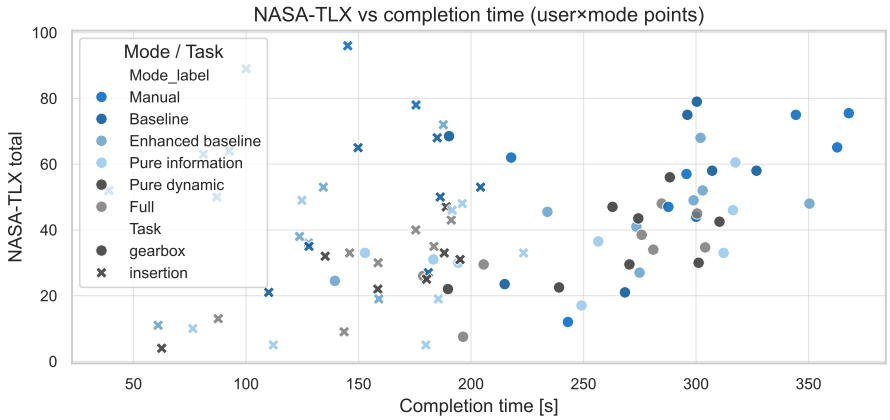


Figure 6.17: NASA–TLX total vs. completion time (user×mode points) for both tasks.

substantially higher workload. This aligns with the earlier observation that dynamic modes, and especially the insertion helper, reduce high-percentile forces and torques. The combined picture suggests a plausible mechanism: by stabilizing contact interaction and damping misalignment, the recommender reduces physically and cognitively stressful events, which is reflected in lower TLX scores.

For completeness, Fig. 6.17 shows TLX vs. completion time across all participant×mode points.

The overall trend is positive, driven mainly by the gearbox task (which showed $r = 0.57$ between TLX and time). However, the dispersion of insertion points along the time axis and the weak correlation for that task confirm that, in contact-dominated scenarios, interaction quality (forces and smoothness) is more strongly tied to workload than raw speed.

In summary, the cross-condition analyses show that the recommender-supported modes consistently reduce workload across tasks, move the operation towards regions of smoother, lower-force interaction, and do so without introducing a systematic penalty in completion time.

6.4 Summary of results

Across both tasks, the results show a consistent benefit of the enhanced feedback modes over Manual and Baseline operation. In the gearbox assembly, the enhanced modes reduce timeouts and shift completion times towards lower values, while clearly lowering contact forces and torques. In the tactile insertion task, which is more contact-heavy and accuracy-dominated, dynamic and informational nudges substantially decrease peak forces and torques and lead to smoother approach and contact motion, without sacrificing execution speed.

Subjective workload follows the same pattern. For both tasks, NASA–TLX overall scores are highest in Manual and Baseline and drop markedly for Enhanced baseline, Pure information, Pure dynamic, and Full. The reduction is driven mainly by mental demand, effort, and frustration, and is more pronounced in the insertion task, where the baseline workload is higher. Cross-condition analyses further indicate that, in insertion, higher peak forces and accelerations correlate strongly with higher workload, supporting the interpretation that smoother, lower-force interaction is experienced as easier and less stressful.

Overall, the recommender-supported modes primarily shift teleoperation into a regime of lower workload and more benign interaction dynamics, with completion times that are comparable to or slightly better than Manual and Baseline. Informational and dynamic nudges both contribute to these effects, and combining them in the Full mode does not incur an additional workload penalty in either task.

7 Discussion

7.1 Overview and main findings

This work investigated how informational and dynamic nudging can be used to support human operators in teleoperation. The central idea was to treat the interface and assistance mechanisms as an artificial recommender system that suggests and modulates actions rather than directly overriding human control. The first research question asked whether such nudging can guide operators towards more effective behavior in concrete teleoperation tasks. The second question concerned longer-term effects: whether the induced guidance has a teaching component that benefits the operator beyond the immediate recommendations.

The experimental results across the gearbox assembly and tactile insertion tasks show a consistent pattern. Compared to a purely manual visual baseline and a conventional haptic baseline, all enhanced feedback modes—ranging from enriched interface information to dynamic guidance and their combination—reduce subjective workload, particularly in terms of mental demand, effort, and frustration. Objective interaction measures indicate that these modes also lead to smoother and more benign contact behavior: peak forces and torques are reduced, and high-acceleration transients become less frequent, especially in the insertion task. Completion times are similar or modestly improved, with a clear reduction in timeouts, but the dominant effect of nudging is an improvement in interaction quality and perceived ease, rather than a dramatic increase in raw speed.

Taken together, these findings support the view that a recommender-style shared control scheme can influence operator behavior in a beneficial way without seizing direct authority over the robot. Informational cues help operators to organize their actions and maintain situational awareness, while dynamic assistance shapes the physical interaction to be more predictable and less stressful. Evidence for long-term teaching effects is naturally limited by the scope and duration of the study, but the stable reduction in workload and contact loads across modes and tasks suggests that the proposed approach is a viable step towards more supportive, human-centered teleoperation systems. The remainder of this chapter discusses these aspects in more detail, relates them to existing work, and reflects on limitations and future directions.

7.2 Interpretation with respect to the research questions

The study addressed two main questions: (i) whether informational nudging can guide operators towards more effective decisions in teleoperation, and (ii) whether such guidance exhibits any teaching effect beyond the immediate recommendations. In the following, we first discuss how the observed changes in behavior and workload relate to the idea of informational guidance,

and then return to potential longer-term effects in Section 7.2.2.

7.2.1 Guiding operators through informational nudging

The first research question asked whether targeted information, delivered through the interface and coordinated by the recommender, can guide operators towards more effective decisions and interaction patterns. In the present context, "effective" is understood in a broad sense: achieving higher success rates and smoother interaction with similar or lower workload, rather than purely minimizing completion time.

Across both tasks, the purely informational mode (Pure information) and the modes that combine enriched information with other enhancements (Enhanced baseline and Full) consistently reduce subjective workload relative to Manual and Baseline. For the gearbox assembly, Pure information lowers the average TLX total from roughly 56 (Manual) and 53 (Baseline) to around 36, with Enhanced baseline and Full in a similar range, despite only modest changes in mean completion time. A comparable pattern appears in the insertion task, where Pure information reduces workload from about 70 (Manual) and 46 (Baseline) to roughly 27, again without a systematic increase in time. This suggests that operators perceive the information-rich modes as substantially less demanding, even when they do not operate significantly faster.

Objective measures show that informational nudging also affects how operators move and interact with the environment. In the gearbox task, informational modes reduce high-percentile contact forces and torques and narrow the distributions of approach accelerations, indicating fewer abrupt corrections and fewer forceful contacts. In the insertion task, where precise alignment is critical, Pure information alone is sufficient to shift contact forces and torques downward and to reduce extreme peaks, even without additional dynamic assistance. This supports the view that the interface cues help users to plan and execute more purposeful approaches, avoiding obviously suboptimal actions such as repeatedly probing at the wrong orientation or targeting implausible parts in the assembly sequence.

The cross-condition comparison further shows that the relative ordering of modes is stable across tasks: Manual and Baseline form a consistently higher-workload, higher-contact baseline, while the information-enhanced modes cluster in a region of lower workload and more benign interaction. This stability across a moderately complex assembly task and a separate, more demanding insertion task suggests that the informational cues do not merely overfit one specific scenario but provide generally useful structure for teleoperation.

At the same time, the information modes do not produce dramatic gains in raw speed, and the correlation between completion time and workload is weak in the insertion task. Informational nudging therefore appears to guide behavior primarily by improving situational awareness, sequence choices, and local alignment strategies, rather than by pushing operators to move faster. In other words, the system helps operators to avoid poor decisions and unnecessary corrections, which manifests as smoother, less stressful interaction and fewer timeouts, but not as aggressive optimization of task duration.

Taken together, these observations support a positive answer to the first research question in a qualified sense. The informational recommender does guide operators towards more effective behavior, as evidenced by reduced workload, gentler contacts, and fewer failures across both

tasks, while leaving direct control authority with the human. The effects are moderate rather than dramatic, and they depend on the informational cues being tightly integrated into an embodied, low-latency interface. Within these constraints, however, informational nudging emerges as a viable and practically attractive way to influence human decision-making in teleoperation.

7.2.2 Teaching and transfer beyond the immediate recommendations

The second research question concerned whether the recommender has a teaching effect that extends beyond the immediate recommendations, i.e. whether operators internalize aspects of the guidance and carry them over to later trials or different modes. In principle, informational and dynamic nudges could shape not only moment-to-moment actions, but also longer-term strategies such as how operators scan the scene, select parts, or align for contact.

Within the scope of the present study, evidence for such transfer is necessarily limited. Trial counts per participant and mode are modest, and the experimental design prioritized breadth of mode comparison over long blocks of practice in a single configuration. Simple analyses of completion time and TLX scores as a function of trial index suggest a weak overall trend towards improved efficiency and slightly lower workload over the course of the experiment, but these changes are small compared to the differences between feedback modes. There is no clear indication that performance in one mode systematically improves after exposure to another specific mode, nor that participants converge to a shared, mode-independent "expert" behavior within the available number of repetitions.

At the same time, some qualitative indications of short-term adaptation can be observed. Participants typically require only a small number of trials to make use of the additional cues in the information and Full modes, and their motion becomes visibly more direct as they learn to interpret the overlays and digital twin. Similarly, once users experience the effect of the insertion helper, they tend to anticipate its behavior and align their own motions accordingly. These observations suggest that operators do not treat the recommender as a black box, but rather incorporate its behavior into their own control strategy on a short time scale.

From a methodological perspective, the present data are therefore better suited to answering the first research question—whether nudging can influence behavior in the moment—than to providing strong claims about teaching and long-term transfer. The results are consistent with the hypothesis that guidance could have a training component: the enhanced modes expose operators to more structured and less noisy interaction patterns, and correlate smoother dynamics with lower workload, which are plausible ingredients for learning. However, demonstrating robust transfer would require longer-term studies with more repetitions, explicit retention tests, or cross-mode evaluation protocols where operators are trained with guidance and then assessed in a reduced or no-assistance condition.

In summary, the experiments provide clear evidence that the recommender can shape behavior and perceived workload during the guided interaction, and tentative indications of short-term adaptation to the provided cues. They do not, by design, permit strong conclusions about lasting teaching effects. The question of how informational and dynamic nudges contribute to durable skill acquisition remains an important direction for future work.

7.3 Mechanisms of influence: information vs dynamics

The experimental results suggest that informational and dynamic nudges influence operator behavior through partially different, but complementary, mechanisms. Informational cues primarily act on how operators plan their actions and allocate attention, while dynamic assistance modifies the physical interaction between human and robot. Their combined effect is most visible in measures of workload and contact quality, particularly in the insertion task.

Informational nudges structure the decision space by making task-relevant relations salient [8, 51]. In the gearbox assembly, highlighting admissible parts, providing sequence-related hints, and visualizing distances or alignment reduces the need for exhaustive visual search and ad-hoc reasoning about what to do next. In the insertion task, target- and wrist-locked overlays and the digital twin provide explicit cues about orientation and progress towards successful insertion. These cues do not force a particular motion, but they influence where operators look and which options they consider plausible. The result is a shift from hesitant, trial-and-error exploration towards more directed approaches and fewer obviously suboptimal attempts. This is reflected in reduced high-percentile accelerations and contact loads in the informational modes, even though the underlying low-level control is unchanged.

Dynamic nudges, in contrast, shape the physical interaction. Guidance fields that attract the end-effector towards the current target, resistive elements that discourage unsafe motions, and the insertion helper all modify the mapping from handle motion to robot motion and contact forces. In free-space motion, these fields regularize trajectories by damping abrupt corrections and providing a "soft funnel" towards task-relevant regions. In contact, they stabilize interaction by aligning the peg with the hole and reducing lateral forces and torques. The strong correlations observed in the insertion task between TLX and the 95th-percentile of contact force ($r \approx 0.62$) and acceleration ($r \approx 0.55$) suggest that these dynamic effects are directly perceived: trials with smoother, lower-force interaction are also reported as less demanding and frustrating. The weaker relationships in the gearbox task are consistent with its less critical contact dynamics.

The Full mode combines both classes of nudges. One might expect that information and dynamics together would simply add up to a strictly better mode. In practice, the improvements appear to saturate: Full typically lies in the same low-workload, low-force region as the best single-modality modes, rather than creating a clearly separate "superior" cluster. This can be interpreted in two ways. On the one hand, it indicates that informational and dynamic nudges are already effective on their own and that the system reaches a regime where additional support yields diminishing returns in the selected metrics. On the other hand, it suggests that operators may be close to a natural performance ceiling for these tasks under the given conditions. Pushing nudging further—by increasing guidance gains or adding more aggressive overlays—would risk degrading performance through over constraint or cognitive overload rather than producing proportional gains.

Importantly, the Full mode does not incur a workload penalty compared to the single-modality enhanced modes. This indicates that the combination of information and dynamics is coherent from the operator's perspective: visual and haptic cues point in compatible directions, the level of autonomy remains limited, and the system does not attempt to override the user's intentions. Within this design space, informational nudging appears to mainly improve planning and situational awareness, while dynamic nudging improves the local interaction physics. Their

joint effect is to move teleoperation into a regime of smoother, less stressful operation without removing the operator from the control loop.

7.4 Relation to shared control and teleoperation literature

The proposed approach sits at the intersection of several strands of work in shared control and teleoperation: classical shared autonomy, haptic guidance and virtual fixtures, and more recent mixed-initiative and recommender-style assistance schemes.

Classical shared control formulations in teleoperation focus on blending user commands with autonomous robot commands, typically at the trajectory or velocity level. Autonomy modules compute a desired motion based on task objectives or obstacle avoidance, and the final command is obtained by weighted combination with the human input. This can yield strong performance improvements, but also raises issues of authority, transparency, and potential conflicts when the autonomous controller steers the robot against the operator's intent. In contrast, the present work deliberately avoids direct trajectory blending. The operator retains full authority over the handle and thus over the commanded pose of the avatar arm. Assistance enters only indirectly, through modulation of impedance parameters and through the presentation of information. This emphasizes a "nudging" perspective: the system shapes the decision context and interaction dynamics, but does not inject hard motion commands.

The dynamic assistance elements are closely related to haptic guidance and virtual fixtures [1, 20], which impose attractive or repulsive fields in task space to guide the operator along desired paths or constrain motion in certain directions. The guidance fields and insertion helper implemented here can be interpreted as adaptive, task-dependent virtual fixtures that are switched and parameterized by the recommender. However, they are intentionally kept relatively soft and are bounded in strength, so that they bias the interaction without fully constraining it. This differs from rigid virtual fixtures that enforce strict geometric constraints and can effectively turn the human into a trigger for pre-defined motions.

On the information side, the work connects to mixed-initiative and interface-centric approaches where the system manages visual cues, highlights relevant objects, or adapts displays based on inferred intent. The digital twin, target-locked overlays, and wrist-mounted indicators pursue a similar goal: to reduce cognitive load and support better choices by making task structure, distances, and progress explicit. The key difference is that these cues are not entirely scripted; their activation and parameters are chosen online by a learning-based recommender that observes task context and interaction history.

Methodologically, the project also bridges model-based and data-driven perspectives on shared control. The initial Stackelberg formulation with an LQG follower and a Bayesian persuasion-inspired leader represents a mathematically elegant instantiation of a hierarchical optimization viewpoint: the system models how a rational follower reacts to information and assistance, and chooses its strategy accordingly. In practice, the modeling gap to real human behavior and the tuning effort made this approach brittle for real-time deployment. The subsequent shift to contextual bandits replaces the explicit follower model by an empirical, online-updated mapping from context to actions. This sacrifices some theoretical structure but gains robustness and direct use of measured task features. Conceptually, the bandit-based recommender can still

be seen as implementing a pragmatic version of the Stackelberg idea: it observes the outcome of its previous "signals" (informational and dynamic nudges) and adapts them to steer the joint human–robot system towards favorable regions of the state space, without requiring a full closed-form model of human decision-making.

In summary, the proposed system can be viewed as a shared control architecture that emphasizes gentle, recommender-like influence rather than hard arbitration of commands. It combines elements of virtual fixtures and adaptive interfaces under a learning-based decision layer, and explores how such a nudging-style shared control compares to more traditional schemes in terms of workload, interaction quality, and operator experience.

7.5 Methodological reflections and design choices

7.5.1 From Stackelberg LQG model to contextual bandits

The development of the recommender followed two distinct stages: an initial model-based formulation as a Stackelberg game with an LQG follower, and the final implementation as a contextual bandit acting on task-level features. This trajectory reflects a trade-off between conceptual elegance and practical robustness.

In the original formulation, the teleoperation channel and the human operator were modeled jointly as a stochastic dynamical system with an LQG controller acting on an EKF belief state. The recommender played the role of a Stackelberg leader: it chose dynamic and informational parameters while anticipating the optimal response of the LQG follower. This yielded a clean mathematical structure. The Gaussian belief representation provided a natural handle on uncertainty, informational nudges could be interpreted as changes in the observation model, and the bi-level optimization formalized the idea that the system reasons about how its signals shape the operator's internal state.

However, turning this into a real-time controller exposed several limitations. First, the behavioral gap between the LQG follower and actual human operators proved substantial. While local motion patterns could be qualitatively similar, the model was highly sensitive to cost weights, noise covariances, and phase segmentation. Parameter settings that were attractive from a control-theoretic perspective did not necessarily correspond to what operators found useful or natural. For example, the model tended to favour variance reduction in state components where humans did not care about precise information in that phase, leading to nudges that were mathematically justified but practically irrelevant. Second, encoding different kinds of nudges and modalities as modifications of the observation model and cost function required many design choices and hand-tuned parameters, which made the system brittle and hard to extend. Third, the computational budget for online simulation of belief dynamics and LQG responses, even on a reduced horizon, left limited room for richer models or additional constraints.

These issues motivated the shift to a contextual bandit formulation. Instead of modeling the human as an explicit optimal controller, the bandit-based recommender treats the human–robot system as a black box that can be probed via actions and observed via rewards. Task-level features—such as current phase, target identity, relative distances, contact indicators, and recent history of interaction forces—are used as context, and the recommender chooses among a discrete

set of candidate nudges. Outcomes are evaluated through simple reward functions derived from performance and interaction measures, and the bandit updates its action preferences online. This design has several practical advantages:

- It operates directly on measurable quantities that are meaningful for the tasks, without requiring a detailed cognitive or motor model of the operator.
- It is robust to modeling errors in the teleoperation dynamics and to inter-individual variability, because it does not assume optimality of the human but simply learns which nudges tend to work better in which contexts.
- It accommodates different nudge types (informational, dynamic) in a uniform framework, as long as their effects can be evaluated via the reward.
- It can exploit offline data to construct informative priors, so that the system does not have to start from a uniform, potentially suboptimal, exploration policy.

Despite this shift, insights from the Stackelberg LQG attempt remained influential for the final design. The explicit modelling of belief dynamics highlighted the importance of uncertainty and phase-dependent priorities, which informed the selection of context features for the bandit (e.g. distance to target, contact state, phase labels). The distinction between informational and dynamic channels in the leader model motivated the separation into two bandits with different action spaces and reward structures. The notion that informational nudges act by shaping belief, and dynamic nudges by shaping interaction dynamics, guided the choice of reward signals (reductions in contact forces, smoother motion, fewer timeouts) as proxies for successful influence.

In hindsight, the model-based Stackelberg formulation can be seen as a conceptual scaffold: it provided a principled vocabulary for thinking about leader–follower interaction and informational influence, but its direct real-time implementation was too fragile for the experimental setting. The contextual bandit approach retains the hierarchical perspective—system actions are chosen with the expectation that they will affect future human behavior—but delegates the details of the human response to data. This combination of a theoretically motivated structure with a pragmatic, data-driven learning method turned out to be more suitable for the targeted teleoperation scenarios.

7.5.2 Teleoperation architecture and embodiment

The teleoperation architecture was deliberately designed to provide a strongly embodied experience: operators see the environment through a head-mounted stereo camera, hear binaural audio from the task space, and interact with the robot through a grounded haptic device that directly drives a human-like arm. This level of embodiment is not incidental. Subtle informational and dynamic nudges are only meaningful if the underlying teleoperation channel is already intuitive and predictable; otherwise, basic usability issues dominate the experience and confound the effects of the recommender .[28, 44, 50].

Several architectural decisions were crucial to achieve this baseline. On the visual side, rendering in VR using head-locked stereo layers and low-latency NDI streaming ensured that the camera motion closely followed head motion, minimizing lag and motion sickness [9, 48]. Task-relevant

overlays and the digital twin were anchored consistently in the same world frame as the robot and objects, so that visual cues aligned geometrically with the physical interaction. The binaural audio stream reinforced the sense of presence by conveying contact and background sounds from the robot workspace without additional artificial processing.

On the control side, Cartesian impedance for the avatar arm provided a natural mapping between handle motion and end-effector motion. Operators could think and act in task space, while the controller absorbed small inconsistencies and contact events as compliant behavior rather than abrupt corrections. The pan–tilt head control, driven by the operator's head pose with appropriate limits and damping, made the viewpoint respond in a human-like way. Together, these choices ensured that the avatar behaved as a physically plausible extension of the operator, which is a prerequisite for interpreting measured changes in motion and forces as responses to nudges rather than artifacts of the control scheme.

The perception backbone and world-frame representation tied these components together. By expressing robot poses, object poses, and visual overlays in a common coordinate system, the system could deliver consistent feedback across modalities: the same object that was highlighted in the digital twin could be targeted by guidance fields and stabilized in haptic interaction. Finally, the central state machine and communication architecture provided reliable coordination between devices while keeping low-level control loops independent and responsive. This separation allowed the recommender to inject informational and dynamic cues at the task level without compromising basic stability or embodiment.

In summary, the architecture was not only a technical platform but also an experimental condition: it established a realistic, embodied teleoperation baseline against which the influence of informational and dynamic nudging could be meaningfully assessed.

7.6 Limitations

The findings of this work should be interpreted in light of several experimental, methodological, and modeling limitations.

From an experimental perspective, the number of participants was relatively small, and each session was long and demanding. Some operators reported mild motion sickness [23, 48] or arm fatigue towards the end of the experiment, which may have affected their performance and workload ratings in later trials. The study was conducted on a single teleoperation platform (arm, haptic device, VR system) in a laboratory environment, so it is unclear to what extent the results transfer to different hardware, tasks, or user populations. The two tasks, while representative of common teleoperation scenarios, also have limitations. The gearbox assembly, in particular, has a modest cognitive demand: after a few initial trials, most participants settled into a stable routine and no longer needed to reason explicitly about the sequence or exploit subtle optimizations, especially given the small workspace and short path lengths. This reduces the room for informational nudges to improve overt performance. Furthermore, the experiment did not include long-term training or retention phases, so the data provide only limited evidence on learning and transfer effects over longer time scales.

Methodologically, the contextual bandit formulation relies on a hand-crafted state and action space and on reward signals derived from available proxies such as completion time, contact

forces, and simple task-level outcomes. In practice, these rewards are noisy and confounded by factors that are difficult to observe or separate, including communication delays, transient operator fatigue, individual differences in risk tolerance, and the operator's momentary ability to interpret the presented information. Given the limited number of repetitions per mode and context, the system has only restricted opportunities to explore and adapt, and it is therefore difficult to claim that the learned policies are close to optimal. The observed improvements should thus be seen as evidence that the chosen nudges can be beneficial, not as proof that the bandit has converged to a best possible strategy. Statistical power is likewise limited: not every apparent trend reaches conventional significance thresholds, and some mode differences are better described as consistent patterns across metrics rather than as formally confirmed effects.

On the modeling side, the representation of operator intent and internal state is deliberately simple. Phase segmentation, intent inference, and target prediction use lightweight models and heuristic features, constrained by the overall scope of the project and the need to maintain real-time performance. These estimates can be inaccurate, especially in ambiguous situations, and misclassifications propagate directly into the recommender's context. As a result, some nudges may be issued based on incorrect assumptions about what the operator is trying to do. More broadly, the system does not include a detailed cognitive model of human decision-making; it captures only coarse aspects of strategy and uncertainty.

Finally, the approach is sensitive to implementation quality, particularly in the interface and assistance channels. The effect of a given piece of information depends critically on where, when, and how it is presented. Poorly placed or visually overloaded overlays may be ignored or misunderstood, and haptic cues that are too weak, too strong, or inconsistent with visual feedback can become distracting or even counterproductive. Considerable engineering effort is required to achieve the level of embodiment and coherence used in this study, and the results implicitly rely on this maturity. In less refined implementations, the same conceptual nudges might show weaker or qualitatively different effects.

Taken together, these limitations imply that the present results should be viewed as indicative rather than definitive. They demonstrate that informational and dynamic nudging can improve workload and interaction quality in the tested scenarios, but they do not fully characterize the space of possible nudging strategies, nor do they establish general performance guarantees across tasks, platforms, and user groups. Further studies with larger samples, richer models, and more diverse conditions are needed to validate and extend the conclusions.

7.7 Implications and future work

The results of this work have several implications for the design of shared-control teleoperation systems. First, low-cost informational nudges already have a clear effect. Modes that only modify the interface—by highlighting admissible parts, visualizing distances and alignment, or focusing the digital twin—consistently reduce workload and contact extremes without changing the low-level controller. This suggests that substantial benefit can be achieved by carefully structuring what information is shown where and when, before introducing more intrusive assistance. In practice, this favors designs that invest in world-consistent overlays, intent-aware

highlighting, and simple progress indicators, provided they are tightly aligned with the operator's viewpoint and task context.

Second, shaping contact dynamics is crucial for perceived workload in contact-dominated tasks. The insertion results show that high-percentile forces and accelerations correlate strongly with TLX, and that dynamic nudges such as guidance fields and insertion helpers effectively reduce these peaks. Designing assistance that explicitly targets the interaction dynamics—by damping misalignment and avoiding abrupt transitions—appears to be a direct route to more comfortable teleoperation. At the same time, the saturation observed between dynamic and Full modes indicates that there is a practical ceiling: beyond a certain level of guidance, further increasing assistance strength or adding more cues is unlikely to yield proportional gains and may risk over-constraint or cognitive overload.

Third, the overall architecture matters. The combination of an embodied VR interface, Cartesian impedance control, a consistent world-frame perception backbone, and a central coordination layer provides a coherent substrate on which nudges can act. In less integrated setups, the same informational or dynamic cues might not align as well with the operator's experience, and their effect could be weaker or more variable.

Several directions for future work follow directly from these observations. On the experimental side, longer-term studies with more participants and extended training phases would be needed to assess teaching and transfer effects more rigorously. This includes protocols where operators are trained with guidance and then evaluated in reduced-assistance or no-assistance modes, as well as retention tests after delays. Expanding the task set to include more complex procedural tasks, cluttered environments, or time-critical scenarios would help to assess how robust the observed patterns are beyond the two scenarios examined here.

On the algorithmic side, the contextual bandit formulation could be extended in several ways. Richer context representations—e.g. including temporal features, eye-tracking data, or more detailed phase and intent estimates—may allow the recommender to distinguish finer-grained situations and adapt nudges more precisely. The reward structure could be refined to integrate multiple objectives, such as explicit safety margins, user preferences, or long-term learning progress. Moving from bandits to lightweight reinforcement-learning methods could allow the recommender to account for delayed effects of its actions, at the cost of more complex credit assignment.

Finally, there is an opportunity to revisit model-based approaches with more data and improved human models. The Stackelberg LQG formulation used here as a conceptual starting point could be combined with learned components, for example by fitting simplified belief and cost models from interaction data instead of specifying them a priori. Hybrid schemes that retain a model-based core for safety and constraint handling, while using data-driven components to capture human variability, may offer a promising compromise between interpretability and robustness. More broadly, integrating nudging-style shared control with higher levels of autonomy—such as task planners or supervisory agents—remains an open avenue for future teleoperation systems that support human operators without displacing them.

8 Conclusion

8.1 Summary of this work

This work explored how informational and dynamic nudging can support human operators in teleoperation without overriding their control authority. Instead of treating shared control as a problem of blending human and autonomous robot commands at the trajectory level, the system was framed as an artificial recommender that selectively exposes information and modulates assistance parameters to bias decisions and interaction in a favorable way.

To investigate this idea, a teleoperation architecture was developed that combines a VR-based visual interface, binaural audio, and a grounded haptic device controlling a human-like robot arm with Cartesian impedance control. A perception pipeline provides object poses in a consistent world frame, and a central coordination layer manages distributed state machines and integrates the recommender. Conceptually, the work started from a Stackelberg formulation with an LQG follower and informational influence modeled via belief updates, but practical limitations led to a contextual bandit recommender that operates directly on task-level features. Two bandits independently selected informational and dynamic nudges during task execution.

The system was evaluated in a user study with two tasks: a gearbox assembly scenario and a tactile insertion scenario. Six feedback modes were compared, ranging from manual visual-only operation to baseline haptics and audio, enhanced interface information, purely dynamic assistance, and a Full mode combining all nudges. Objective measures (success rate, completion time, kinematic features, contact forces and torques) and subjective workload (NASA–TLX) were collected.

Across both tasks, the enhanced modes consistently reduced workload, especially mental demand, effort, and frustration, and decreased peak contact loads and high-acceleration transients, with completion times that were similar to or modestly better than the baselines. These results support the view that recommender-style nudging can improve interaction quality and operator experience without removing the human from the control loop.

8.2 Answers to the research questions

RQ1: Can we guide a human operator towards more optimal decisions leveraging informational nudging? Within the scope of the tested tasks, the results indicate a positive answer in a qualified sense. Informationally enhanced modes (Enhanced baseline, Pure information, and Full) consistently reduced subjective workload relative to Manual and Baseline and led to smoother interaction, as evidenced by lower high-percentile forces and accelerations and fewer extreme contact events. In the gearbox task, these improvements came with modest reductions

in timeouts and a shift towards shorter completion times. In the insertion task, informational cues alone were sufficient to reduce peak forces and workload substantially, even without additional dynamic assistance, while maintaining comparable average times. Informational nudging therefore guided operators towards more effective behavior in terms of comfort, contact quality, and robustness to time pressure, even if it did not aggressively optimize raw speed.

RQ2: Can we make a teaching impact to the human operator beyond the recommender environment? Evidence for longer-term teaching effects is more limited. Participants adapted quickly to the provided cues within a few trials, and their motion became more direct and regular as they learned how to interpret overlays, the digital twin, and the insertion helper. However, the available data do not show strong, mode-independent improvements that would clearly indicate lasting transfer beyond the immediate recommendations. Trial counts per participant and mode were modest, and there was no explicit retention or post-training evaluation without assistance. The experiments therefore demonstrate short-term adaptation to the guidance but do not allow firm conclusions about durable skill transfer. The results are consistent with the hypothesis that teaching effects could emerge in longer-term use, but this remains an open question for future work.

8.3 Contributions

The main contributions of this thesis can be grouped into conceptual, technical, and empirical aspects.

Conceptual contributions

- A reframing of shared control in teleoperation as a nudging or recommender problem, where the system shapes information and assistance parameters rather than directly blending human and autonomous commands.
- A clear distinction between informational and dynamic nudges as two complementary channels for influencing operator behavior: information shapes planning, attention, and beliefs; dynamics shape the physical interaction and contact behaviour.
- An exploration of model-based and data-driven perspectives on informational influence, starting from a Stackelberg LQG formulation and arriving at a contextual bandit implementation that retains the hierarchical viewpoint while relaxing modelling assumptions.

Technical contributions

- The design and implementation of a teleoperation architecture with strong embodiment, combining:
  - a VR interface with low-latency stereo video, head-locked HUD elements, target- and wrist-locked overlays, and an integrated digital twin,

  - haptic teleoperation of a human-like Franka avatar via Cartesian impedance control with null-space posture regulation,
  - pan–tilt head control driven by operator head motion and a perception pipeline that estimates object poses in a consistent world frame,
  - a central coordination layer with distributed state machines and UDP-based communication.
- The implementation of a two-bandit recommender:
  - a dynamic bandit that selects assistance parameters (guidance gains, resistance levels, insertion helper),
  - an informational bandit that activates and parametrizes visual, haptic, and audio cues,
  - context features derived from intent inference, sequence planning, task phase, and interaction statistics,
  - priors constructed from a deterministic policy learned in offline experiments.

Empirical contributions

- A controlled user study comparing six feedback modes across two teleoperation tasks with different characteristics (mixed reaching/placement vs. precision insertion).
- Empirical evidence that enhanced informational and dynamic modes:
  - reduce subjective workload, particularly mental demand, effort, and frustration, in both tasks,
  - reduce peak contact forces and torques and smooth motion, especially in the contact-rich insertion task,
  - achieve these improvements without systematic penalties in completion time and with fewer timeouts.
- An analysis of cross-condition relationships showing that, in insertion, high-percentile forces and accelerations correlate strongly with workload, supporting the interpretation that smoother interaction is directly perceived as easier.

8.4 Outlook

Several avenues for future research follow from this work.

On the experimental side, larger-scale and longer-term studies are needed to characterize teaching and transfer effects more rigorously. This includes protocols where operators are explicitly trained with guidance and then evaluated under reduced or no assistance, as well as retention tests after delays. Extending the evaluation to more diverse tasks—such as cluttered manipulation, remote inspection, or time-critical interventions—would help to assess how robust the observed benefits are across domains.

On the algorithmic side, the contextual bandit recommender can be refined and extended. Richer context representations that incorporate temporal features, eye-tracking or gaze proxies, or more detailed intent estimates could allow the system to distinguish finer-grained situations and adapt nudges more precisely. Reward functions could be broadened to include explicit safety margins, user preferences, or measures of learning progress. Moving towards lightweight reinforcement learning may enable the system to account for delayed consequences of nudges while maintaining interpretability and safety constraints.

Finally, there is potential in hybrid approaches that combine model-based structure with data-driven components. Insights from the initial Stackelberg LQG formulation could be revisited using human-in-the-loop data to fit simplified belief and cost models, or to design safety envelopes within which learning-based nudges operate. More generally, integrating nudging-style shared control with higher-level task planners and supervisory autonomy offers a route towards teleoperation systems that are supportive by default: they preserve human authority over actions while actively shaping information and interaction dynamics to make those actions more effective and less demanding.

Overall, the results of this thesis suggest that treating shared control as a problem of recommendation and nudging is a promising direction. With further development of human models, learning methods, and experimental validation, such approaches may contribute to teleoperation systems that are not only more capable, but also more comfortable and sustainable for their human operators.

9 References

[1] Jake J. Abbott, Panadda Marayong, and Allison M. Okamura. "Haptic Virtual Fixtures for Robot-Assisted Manipulation". In: *International Symposium of Robotics Research (ISRR)*. 2007. URL: https://robots.stanford.edu/isrr-papers/final/final-06.pdf.

[2] Jake J. Abbott and Allison M. Okamura. "Stable Forbidden-Region Virtual Fixtures for Bilateral Telemanipulation". In: *Journal of Dynamic Systems, Measurement, and Control* 128.1 (2006), pp. 53–64. DOI: 10.1115/1.2168163.

[3] Shipra Agrawal and Navin Goyal. "Thompson Sampling for Contextual Bandits with Linear Payoffs". In: *Proceedings of the 30th International Conference on Machine Learning (ICML)*. 2013. URL: https://proceedings.mlr.press/v28/agrawal13.html.

[4] Shipra Agrawal and Navin Goyal. "Thompson Sampling for Contextual Bandits with Linear Payoffs". In: *Proceedings of the 30th International Conference on Machine Learning (ICML)*. Vol. 28. Proceedings of Machine Learning Research. 2013, pp. 127–135.

[5] Anonymous. "Simon's Bounded Rationality". In: *Decisions in Economics and Finance* 47.2 (2024), pp. 327–346.

[6] Steven A. Bowyer, Brian L. Davies, and Francisco R. y Baena. "Active Constraints/Virtual Fixtures: A Survey". In: *IEEE Transactions on Robotics* 30.1 (2014), pp. 138–157. DOI: 10.1109/TRO.2013.2283410.

[7] Gary Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[8] Ana Caraban et al. "23 Ways to Nudge: A Review of Technology-Mediated Nudging in Human-Computer Interaction". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. 2019, pp. 1–15. DOI: 10.1145/3290605.3300733.

[9] E. Chang, H. T. Kim, and B. Yoo. "Virtual Reality Sickness: A Review of Causes and Measurements". In: *International Journal of Human–Computer Interaction* 36.17 (2020), pp. 1658–1682. DOI: 10.1080/10447318.2020.1778351.

[10] Andre Coelho et al. "Whole-Body Teleoperation and Shared Control of Redundant Robots with Applications to Aerial Manipulation". In: *Journal of Intelligent & Robotic Systems* 102.14 (2021), –. DOI: 10.1007/s10846-021-01365-7.

[11] Erwin Coumans. "Bullet Physics Simulation". In: *ACM SIGGRAPH 2015 Courses* (2015).

[12] Kourosh Darvish et al. "Teleoperation of Humanoid Robots: A Survey". In: *IEEE Transactions on Robotics* 39.3 (2023), pp. 1706–1727. DOI: 10.1109/TRO.2023.3236952.

[13] Mica R. Endsley. "Toward a Theory of Situation Awareness in Dynamic Systems". In: *Human Factors* 37.1 (1995), pp. 32–64. DOI: 10.1518/001872095779049543.

[14] Kazem Falahati. "The Standard Model of Rational Risky Decision-Making". In: *Journal of Risk and Financial Management* 14.4 (2021), p. 158. DOI: 10.3390/jrfm14040158.

[15] Force Dimension. *sigma.7 Haptic Device — Specification Sheet*. https://www.forcedimension.com/images/doc/specsheet_-_sigma7.pdf. Accessed: 2025-12-13.

[16] Franka Emika GmbH. *libfranka: A C++ Library for the Franka Emika Panda Robot*. Accessed: 2025-03-08. 2023. URL: https://github.com/frankaemika/libfranka.

[17] Gerd Gigerenzer et al. "Towards a Rational Theory of Heuristics". In: *Minds, Models and Milieux*. 2005, pp. 34–59.

[18] Sandra G. Hart and Lowell E. Staveland. “Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research”. In: *Human Mental Workload*. Vol. 52. Advances in Psychology. North-Holland, 1988, pp. 139–183. DOI: 10.1016/S0166-4115(08)62386-9.

[19] Neville Hogan. “Impedance Control: An Approach to Manipulation: Part I—Theory”. In: *Journal of Dynamic Systems, Measurement, and Control* 107.1 (1985), pp. 1–7. DOI: 10.1115/1.3140702.

[20] Neville Hogan. “Impedance Control: An Approach to Manipulation: Part I—Theory”. In: *Journal of Dynamic Systems, Measurement, and Control* 107.1 (1985), pp. 1–7.

[21] Yuhan Hu et al. “Nudging or Waiting?: Automatically Synthesized Robot Strategies for Evacuating Noncompliant Users in an Emergency Situation”. In: *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2023, pp. 603–611.

[22] Siddarth Jain and Brenna Argall. “Probabilistic Human Intent Recognition for Shared Autonomy in Assistive Robotics”. In: *ACM Transactions on Human-Robot Interaction* 9.1 (2019), p. 2. DOI: 10.1145/3359614.

[23] Robert S. Kennedy et al. “Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness”. In: *The International Journal of Aviation Psychology* 3.3 (1993), pp. 203–220. DOI: 10.1207/s15327108ijap0303_3.

[24] Oussama Khatib. “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots”. In: *The International Journal of Robotics Research* 5.1 (1986), pp. 90–98. DOI: 10.1177/027836498600500106.

[25] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020. ISBN: 9781108486828.

[26] Joseph J. LaViola. “A Discussion of Cybersickness in Virtual Environments”. In: *ACM SIGCHI Bulletin* 32.1 (2000), pp. 47–56. DOI: 10.1145/333329.333344.

[27] John D. Lee and Katrina A. See. “Trust in Automation: Designing for Appropriate Reliance”. In: *Human Factors* 46.1 (2004), pp. 50–80. DOI: 10.1518/hfes.46.1.50_30392.

[28] John D. Lee and Katrina A. See. “Trust in Automation: Designing for Appropriate Reliance”. In: *Human Factors* 46.1 (2004), pp. 50–80. DOI: 10.1518/hfes.46.1.50_30392.

[29] Thomas C Leonard. *Richard H. Thaler, Cass R. Sunstein, Nudge: Improving Decisions about Health, Wealth, and Happiness: Yale University Press, New Haven, CT, 2008, 293 pp*, 26.00. 2008.

[30] Gaofeng Li et al. “The Classification and New Trends of Shared Control Strategies in Telerobotic Systems: A Survey”. In: *IEEE Transactions on Haptics* 16.2 (2023), pp. 118–133. DOI: 10.1109/TOH.2023.3253856.

[31] Lihong Li et al. “A Contextual-Bandit Approach to Personalized News Article Recommendation”. In: *Proceedings of the 19th International Conference on World Wide Web (WWW)*. 2010, pp. 661–670. DOI: 10.1145/1772690.1772758.

[32] Lihong Li et al. “A Contextual-Bandit Approach to Personalized News Article Recommendation”. In: *Proceedings of the 19th International Conference on World Wide Web (WWW ’10)*. 2010, pp. 661–670. DOI: 10.1145/1772690.1772758.

[33] David A Lieberman. *Learning: Behavior and cognition*. Thomson Brooks/Cole Publishing Co, 1993.

[34] Dylan P Losey et al. “A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction”. In: *Applied Mechanics Reviews* 70.1 (2018), p. 010804.

[35] Linyuan Lü et al. “Recommender systems”. In: *Physics reports* 519.1 (2012), pp. 1–49.

[36] MessagePack Developers. *MessagePack Specification*. https://github.com/msgpack/msgpack/blob/master/spec.md. Accessed: 2025-12-13. 2017.

[37] *NASA Task Load Index (TLX): Paper and Pencil Package*. Tech. rep. NASA Ames Research Center, 1986. URL: https://ntrs.nasa.gov/api/citations/20000021488/downloads/20000021488.pdf.

[38] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[39] Gunter Niemeyer and Jean-Jacques E. Slotine. "Telemanipulation with Time Delays". In: *The International Journal of Robotics Research* 23.9 (2004), pp. 873–890. DOI: 10.1177/0278364904045563.

[40] Stefanos Nikolaidis and Julie A. Shah. "Human–Robot Mutual Adaptation in Shared Autonomy". In: *2015 ACM/IEEE International Conference on Human–Robot Interaction (HRI)*. 2015, pp. 91–98. DOI: 10.1145/2696454.2696468.

[41] Edwin Olson. "AprilTag: A Robust and Flexible Visual Fiducial System". In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*. 2011, pp. 3400–3407. DOI: 10.1109/ICRA.2011.5979561.

[42] Dimitris Panagopoulos et al. "Intent Inference in Shared-Control Teleoperation System in Consideration of User Behavior". In: *Complex & Intelligent Systems* 8.4 (2022), pp. 2971–2981. DOI: 10.1007/s40747-021-00533-4.

[43] Raja Parasuraman, Thomas B. Sheridan, and Christopher D. Wickens. "A Model for Types and Levels of Human Interaction with Automation". In: *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans* 30.3 (2000), pp. 286–297. DOI: 10.1109/3468.844354.

[44] Raja Parasuraman, Thomas B. Sheridan, and Christopher D. Wickens. "A Model for Types and Levels of Human Interaction with Automation". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans* 30.3 (2000), pp. 286–297. DOI: 10.1109/3468.844354.

[45] Louis B. Rosenberg. "Virtual Fixtures as Tools to Enhance Operator Performance in Telepresence Environments". In: *Proc. SPIE 2057, Telemanipulator Technology and Space Telerobotics*. 1993. DOI: 10.1117/12.164901.

[46] L. Rozo et al. "Learning controllers for reactive and proactive behaviors in human–robot collaboration". In: *Frontiers in Neurorobotics* 7 (2013), pp. 1–15. DOI: 10.3389/fnbot.2013.00001.

[47] Daniel Russo and Benjamin Van Roy. "Learning to Optimize via Posterior Sampling". In: *Mathematics of Operations Research* 39.4 (2014), pp. 1221–1243. DOI: 10.1287/moor.2014.0650.

[48] Dimitrios Saredakis et al. "Factors Associated With Virtual Reality Sickness in Head-Mounted Displays: A Systematic Review and Meta-Analysis". In: *Frontiers in Human Neuroscience* 14 (2020), p. 96. DOI: 10.3389/fnhum.2020.00096.

[49] Leonard J. Savage. *The Foundations of Statistics*. John Wiley & Sons, 1954.

[50] Thomas B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. Cambridge, MA: MIT Press, 1992.

[51] Richard H. Thaler and Cass R. Sunstein. *Nudge: Improving Decisions About Health, Wealth, and Happiness*. New Haven, CT: Yale University Press, 2008. ISBN: 9780300122237.

[52] Emanuel Todorov, Tom Erez, and Yuval Tassa. "MuJoCo: A Physics Engine for Model-Based Control". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2012), pp. 5026–5032. DOI: 10.1109/IROS.2012.6386109.

[53] Evangelos Tsagkournis et al. "A Supervised Machine Learning Approach to Operator Intent Recognition for Teleoperated Mobile Robot Navigation". In: *IFAC–PapersOnLine* 56.31 (2023), 10.1016/j.ifacol.2023.10.1023. DOI: 10.1016/j.ifacol.2023.10.1023.

[54] Séamas Weech, Sophie Kenny, and Michael Barnett-Cowan. "Presence and Cybersickness in Virtual Reality Are Negatively Related: A Review". In: *Frontiers in Psychology* 10 (2019), p. 158. DOI: 10.3389/fpsyg.2019.00158.

[55] Daniel E. Whitney. "Quasi-Static Assembly of Compliantly Supported Rigid Parts". In: *Journal of Dynamic Systems, Measurement, and Control* 104.1 (1982), pp. 65–77. DOI: 10.1115/1.3149637.

[56] Zhengyou Zhang. "A Flexible New Technique for Camera Calibration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11 (2000), pp. 1330–1334. DOI: 10.1109/34.888718.