"Current developments in programming through demonstration and augmented reality-A Review"

Devaraj E[*1], Venkatesh S N[2], Raghavendra H[3], J R Jessy Michla[4]

[*1] Assistant Professor, Department of Mechanical Engg, CMR University, Banaglore, Karnataka
[2], Assistant professor, Electronics and telecommunication engineering, Dr. Ambedkar Institute of Technology, Bangalore, Karnataka
[3], Assistant Professor, Department of Mechanical Engg, Madadapalle Institute of Technology & Science, Madanapalle, AP
[4], Associate Professor, Department of Aeronautical Engineering, Kalasalingam Academy of Research and Education, Virudhunagar

Abstract

The demand for complex robotic systems is rising along with the need for new instruments and techniques to boost the effectiveness and efficiency of robot programming. Effective task communication between the human and the robot is essential for the latter. Two rapidly emerging technologies, augmented reality (AR) and programming by demonstration (PbD), have garnered significant attention in recent years. PbD allows users to teach a robotic system by example, eliminating the need for explicit programming to accomplish a task. In contrast, augmented reality (AR) overlays digital information over real-world objects to produce an interactive experience. Both technologies are being researched for a variety of applications, such as industrial automation and healthcare, and show a lot of potential. The groundwork has been completed for examining the areas of convergence where merging these two methodologies could offer advantageous viewpoints on how to address the problem of intuitive robot programming and, by extension, human-robot interaction (HRI).

1.0 Overview of Robot Programming via Illustration

Robot programming is a basic component of the many subfields of Human-Robot Interaction (HRI), a rapidly expanding field of study. The primary objective is to comprehend, create, and assess robotic systems that are capable of safe, effective, and intuitive human-robot interaction. An essential part of this strategy are collaborative robots, which are made to work side by side with people in shared workspaces to increase efficiency and production. In the perspective of reintroducing Industry 4.0, the current wave of the industrial revolution (Figure 1), For this phase to succeed, new approaches to the challenge of developing more flexible and intuitive methodologies and procedures for robot programming are required. That being said, one possible path of action would be to develop Humancentered Robotics (HCR) collaborative environments with better HRI measurements in order to characterize the previously described aims.

The latter could be achieved through the local industry ecosystem's integration of various cutting edge technology. The primary issue with traditional robot programming techniques, such as teaching pedant, offline programming, etc., is that they require sophisticated user understanding of high-level programming, which can be challenging to obtain. Furthermore, when they are used, several of the previously described techniques may be risky and ineffective. Robot Programming by Demonstration (PbD) is a relatively new programming technique that is still being researched. It is simpler to learn and has the potential to enable more precise and efficient programming.

It still has drawbacks, though, in its current configuration. Its limited applicability stems from the requirement that the robot be able to identify and comprehend the task being demonstrated. In order to accomplish this, it can occur to employ contemporary data-driven exploration approaches to teach the robot new abilities and how to perform certain tasks. Additionally, scripting languages and graphical programming languages can be utilized to bring a more visual programming environment and produce faster results in a more convenient method. Although this method has a more efficient positive impact on the robot's learning process outcome, it leaves the user with an incomplete impression of the system. As such, it has a detrimental effect on the desired result and does not enhance the process as a whole. This results from the information gap that exists between the user (teacher) and the robot (learner), one of the primary PbD problems. Even though the user is very knowledgeable about the task being taught, it's possible that they lack the technical know-how to program the robot. However, even while the robot is technically capable of completing the work, it might not comprehend the fundamental ideas that the user is attempting to apply.

Adding another quickly developing technology, augmented reality (AR) (also known as mixed reality, or MR for interactive frameworks that allow virtual objects to interact with the real world, creating a seamless hybrid environment) through the use of head-mounted displays (HMD), may be able to solve that issue. Robots may learn complex jobs more quickly and accurately with PbD that uses augmented reality techniques. Additionally, PbD can be used to create new processes that can be tailored to specific conditions and restrictions, increasing procedure productivity and effectiveness. Thus, through virtual item augmentations and immersive experiences, AR and MR are perfect for augmenting the user's physical reality and surrounds. They may also be highly advantageously utilized in the HRC domain.

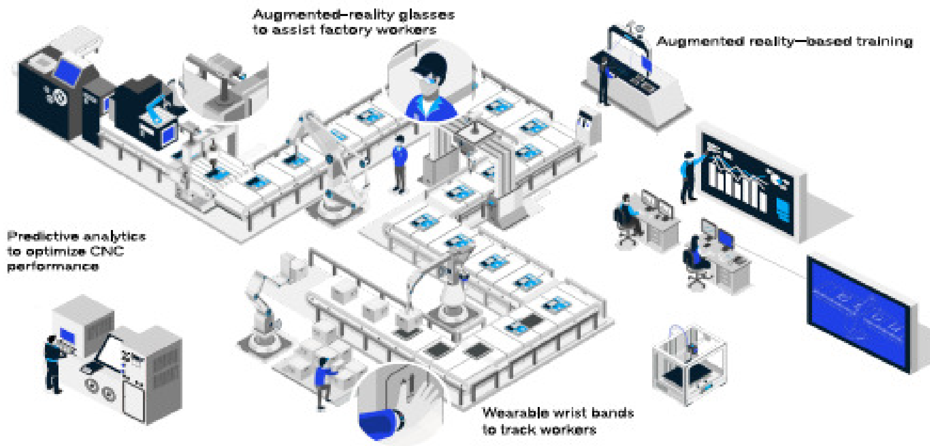

Figure 1: Rendering of the proposed Industry 4.0 environment

In order to improve robot programming performance, a variety of approaches are employed, including the use of Augmented Reality tools and methods. According to the authors of those studies, the approaches' intuitiveness can improve performance metrics for both experts and non-experts. It is clear that, in addition to what the performance indices indicate, the primary goal of any research conducted in this area should be to develop a more immersive interface that would not only make the user feel more in control and benefit from it, but also pique their interest and desire to use it.

Moreover, remote assistance and troubleshooting using AR-enhanced PbD enables quicker problem-solving and more effective maintenance practices. Because the user interacts with holograms during the entire process, integrating PbD and AR techniques can enhance the user experience (UX) and make it more immersive and interactive. This can also increase system safety and dependability.

Within this framework, augmented reality (AR) can aid in closing the previously described knowledge gap by offering a user-friendly and interactive interface that enables more efficient and natural communication between the user and the robot. Using augmented reality (AR), the user can show the robot how to complete a task with physical gestures and actions that can be recorded by cameras and other sensors and subsequently mapped to the robot's control system. In this manner, the user can instruct the robot without needing to create intricate code, and the robot can pick up knowledge from the user's examples. The paradigm known as Robot Programming by Demonstration (PbD) or Learning from Demonstration (LfD)—also referred to as Imitation Learning—allows robots to pick up new abilities by mimicking an expert. PbD is a type of robot programming where a task is demonstrated to the robot in order to educate it how to accomplish it. Within this framework, the user is spared from having to write complex, all-purpose controllers that would allow the robot to change its behavior depending on the situation. A significant amount of human thinking and knowledge implementation would be needed to construct such a controller, which would also need to incorporate a very complex error management algorithm and a costly, non-pragmatic notion. Following a sequence of demonstrations, the robot mimics the behavior it has learned and, when prompted, repeats the task on its own.

The fundamental characteristic of imitation learning is the innate capacity of sophisticated multicellular organisms, such as mammals, to mimic specific behaviors as a result of low-level evolutionary encoding. Its emergence stems from the bioinspired imitation process that both humans and animals employ to acquire new skills. Imitation is frequently seen in early life in both the human and animal kingdoms, either as a natural survival strategy or as a result of social engagement and information sharing with other members of the same or different species. Two categories of biologically inspired imitation systems exist: conceptual models, in which the system attempts to emulate the behavior of its biological equivalent, and connectionist models, in which a low-level artificial neural network is employed. This behavior can be put into user-friendly robotics interfaces that even inexperienced users with no programming background can use. The following are some crucial queries that could be used to understand imitation learning in a more advanced manner:

• Who to imitate? That question refers to the problem of who is the expert in the case of Human Robot Interaction.

• When to imitate? What is the timespan of the imitation and when is the best time for it to take place.

• What to imitate? What is the goal of performing a certain skill.

• How to map? The translation from a demonstration to actual imitating behaviour is not straightforward and usually some encoding procedure of the acquired information has to take place.

• How to evaluate an imitation? A question that concerns implementation of self-improvement techniques for the learnt skills by defining and using the proper metrics.

These issues are crucial to understanding how to approach the subject of programming intuitive robots by demonstration, so they should be carefully looked into in any relevant study that is conducted.

2.0 Skill Encoding and Generalisation

By using PbD, non-experts can teach the robot new tasks in an intuitive way by using several learning modalities that are tailored to the specific situation at hand. Every work can be broken down into discrete talents that the robot can see. The intended behavior can be encoded by repeatedly imitating those skills in different contexts. This is completed when learning a skill during the skill encoding phase. The generalization of the behaviors that were previously learned through examples is the other crucial stage.

In this stage, the robot can comprehend the minor modifications to the assignment and adjust accordingly. This section's primary focus is on these two ideas, which are examined below. One of the primary PbD issues is the skill encoding process, which determines how the robot represents and interprets the learned knowledge. Either a low-level, or non-linear mapping between sensory and motor information, or a high-level, or skill decomposition of action perception units that symbolically code predefined motion elements, can be used to represent a skill. In the former case, the encoded information—a specific movement—provides part of the answer to the central question of what should be emulated.

These representations are called trajectory-based approaches and symbolic approaches, respectively, and are also shown in Figure 2. To elaborate, the symbolic level of encoding, sometimes referred to as the task level, uses a sequential form of skills portrayed in a symbolic style that can be recognized using classification techniques and utilized to establish a hierarchical structure of sorts. These structures are often tree-like and reflect certain state-actions as relations and nodes, respectively.

The robot maps that sequence to its own action set upon skill reproduction. Conversely, the trajectory level pertains to the process of utilizing statistical modeling and dynamical models to map the exhibited trajectory to the robotic one, which is then translated into new robot actions or primitives.

Additional layers of encoding imitation include the model-based level, which employs an exploration-based learning model, and the goal-level method, which uses an inference process to determine the aim after witnessing a certain sequence of activities. In an attempt to combine the advantages of both approaches, a combination of the symbolic level and trajectory level has also been used. This involves syncing sensor data with movement primitives to produce the previously described Associative Skill Memories (ASM) idea. These are built on a limited number of stereotyped movement primitives, but they are sufficient to build up huge manipulation skill-sets, mimicking the way the human brain processes different sensory inputs and performs auto-associative memory. Additionally, they propose that the system can encode a state machine with imposed constraints in the manipulation sequence by incorporating a manipulation graph.

The generalization of a skill is a crucial stage of learning in PbD. The ability of a robot to adjust and apply a taught skill to novel circumstances, task requirements, or surroundings that were not experienced during the initial demonstrations is known as generalization. Stated differently, generalization refers to the process of applying previously learned information to execute a skill in a variety of situations.

In PbD, generalization aims to increase the robustness and flexibility of learnt behaviors by allowing robots to deal with unfamiliar situations without needing explicit demonstrations for every case. In order to generalize symbolic level encoding, a predetermined set of fundamental controllers must be provided in order for the sequential structuring of specified action elements to be learned. This allows for the learning of hierarchy, rules, and loops.

Conversely, at the trajectory encoding level, the generalization span encompasses the generalization of movements. Averaged trajectories and associated variations could be used to depict motion in a generic manner, which would be helpful. The benefit is the ability to encode a wide variety of signals or gestures; the disadvantage is the inability to replicate complicated, high-level skills. Depending on the representation used to encode the talent, there are several ways to achieve generalization. Several methods consist of:

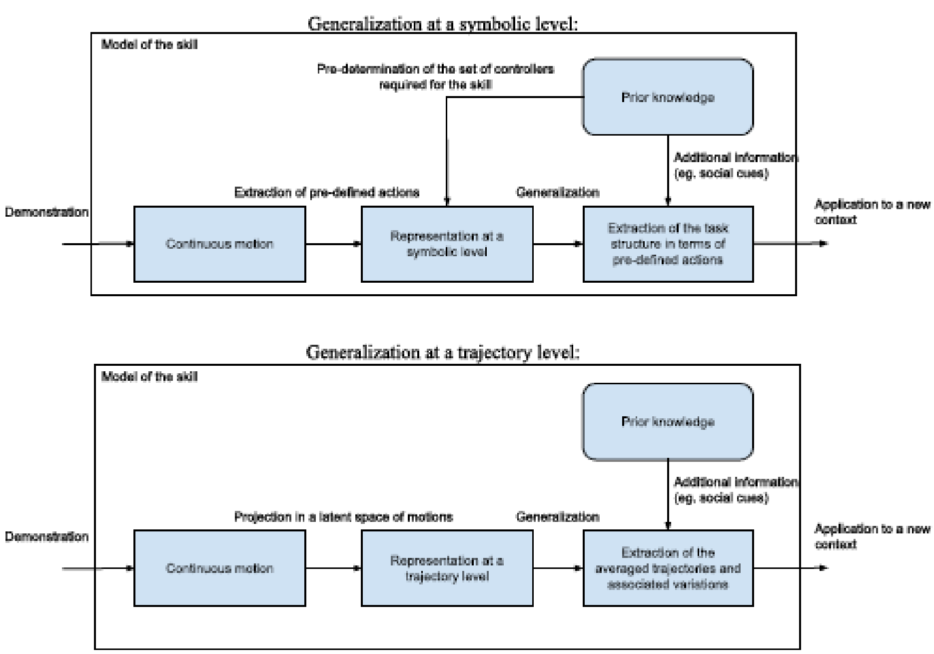

Figure 2: Different levels of representation for describing the skill

Dynamic Movement Primitives (DMPs): By altering the goal position, the movement time, or the movement amplitude, DMPs can be made to adapt to new task conditions. This enables the robot to execute the skill in many situations.

• Gaussian Mixture Models (GMMs) and Gaussian Mixture Regression (GMR): The joint probability distribution of the trajectories that have been illustrated and the associated task parameters can be modeled using GMMs. The most likely trajectory for a given set of job parameters can then be estimated using GMR, which makes it easier to generalize to new circumstances.

• Neural Networks (NN): When a skill is learned using NN, the network can be trained to anticipate the right course of action or set of steps for various task circumstances or surroundings. Once trained, by modifying its internal weights according to the input conditions, the network may generalize to new scenarios.

• Reinforcement Learning (RL): By combining PbD with RL, the robot can use trial and error to hone and adjust the skills it has learnt. This helps the robot operate better under different circumstances and generalize its behavior.

Using the recently released Dynamic Movement Primitives (DMPs) is a common generalization technique. By breaking down complex motor movements into their temporal (canonical system) and spatial (transformation system) components, DMPs are able to depict them. After observing a skill being performed, the robot extracts the trajectory and fits basis functions to the trajectory to learn a DMP. By altering the objective position, the movement time, or the movement amplitude, the learnt DMP can be adjusted to fit various settings or work situations. Gaussian Processes (GP) have expanded DMP's capabilities in a number of ways. More precisely, the human demonstration's intrinsic uncertainty factor—which can occasionally be inaccurate—is being effectively reduced by the generalization for the trajectory encoding situation. In studies by Chernova et al., the previously described issue is solved by presenting an alternative way to express the learning strategy as a collection of GMMs, with each model having numerous GMM components that each correspond to a single action. According to their methodology, the demonstration examples that were gradually received—ideally in a manner that minimizes the quantity of demonstrations required to obtain the policy—served as the training data for the GMMs set.

3.0 Input Modalities and Demonstration Procedure

As mentioned at the beginning of this section, there are several ways to carry out the demonstration procedure. The key to establishing a form is the modality that is employed to gather the data required to build the knowledge base. A variety of sensor inputs and the development and use of multiple interfaces can be used to retrieve information. These methods serve as a semantic translation, enabling the user to comprehend and make appropriate use of the information base. In PbD, "input modalities" refers to the various ways in which the user can show the robot how to perform the intended task. These modalities can include vocal instructions or physical actions like gestures or motions.

In PbD, an efficient interface design is essential since it directly affects the user's capacity to convey the intended task to the robot. A well-designed interface should be simple to use, intuitive, and give the user immediate feedback. PbD allows for the use of a variety of interface types, such as:

Immersive Interfaces: In the case of VR/AR interfaces, these technologies are used to build a virtual environment that can imitate real-world situations. However, by permitting numerous levels of functionality, augmented reality (AR) interfaces employ computer-generated pictures and data to enhance the user's perception of the real environment, thereby offering an interactive and user-friendly interface for the user to explain the task to the robot. More naturally and intuitively, users can interact with virtual items and show the robot duties. While AR interfaces can be particularly effective in PbD because they allow the user to interact with the robot in a more natural and immersive way while also interacting with the physical world, VR interfaces are particularly useful for tasks that require a high degree of precision or for tasks that are difficult to simulate in the real world.

Figure 3: The GUI used for FRANKA EMIKA products that utilizes a workflow-based robot programming approach [EMI].

Graphical User Interfaces (GUIs): These interfaces allow the user to interact with the robot through the use of visual representations like buttons or menus. Because they give users a comfortable and familiar interface, GUIs can be useful in PbD. One prominent illustration is the GUI for FRANKA EMIKA, which is shown in Figure3 and allows the use of numerous processes with a single Application.

Natural Language Interfaces (NLIs): NLIs allow a user to speak with a robot in natural language by utilizing natural language processing (NLP) techniques. The goal of natural language interfaces for robot control is to determine the optimal series of movements that correspond to the desired behavior specified by the command. For users who might lack the technical know-how to use more sophisticated interfaces, NLIs can be helpful.

Voice User Interfaces (VUIs): Also known as Spoken Dialogue Interfaces (SDIs), VUIs make better use of human-machine interfaces by utilizing spoken language, which is the most powerful and natural form of human-to-human communication.

The type of Spoken Dialogue Interface (SDI) used determines how flexible and robust the system is while processing spoken input and output. As such, Spoken Dialogue Systems (SDSs) can be as simple as finite-state systems handling a limited set of commands, or they can be as complex as more advanced systems capable of inference and planning as part of a collaborative interaction approach. VUIs and NLIs can be distinguished primarily by their modalities; the latter can support speech input as well as text-based communication.

Tangible User Interfaces: To illustrate the task being taught, tangible objects like toys or blocks are used in Tangible User Interfaces (TUI). These blocks serve to specify activities and their order as well as to offer annotations for objects, places, or regions. In contrast, the robot makes use of these blocks to recognize items and other blocks in its workspace and arranges them according to instructions by overcoming any related constraints. These manipulable items give users a tactile interface while demonstrating the desired task. Users who prefer a more hands-on approach or who may have little technical experience will find tangible interfaces especially helpful.

The impacts of a tangible system are compared to a group of kids of different ages that have to use a GUI to follow directions for a robot programming activity. The findings suggest that the tangible system is not the most user-friendly interface paradigm and is more difficult for older kids who are accustomed to computers to utilize. For both older and younger learners, it is also more entertaining as a medium. The TUIs were simpler to use for younger kids who aren't accustomed to interacting with computers. In this instance, physical blocks were applied with the TUI without a projected table. A TUI example is shown in Figure 4.

Brain-Computer Interfaces (BCIs): BCIs monitor brain activity using electroencephalography (EEG) or other methods and convert it into orders that the robot can comprehend. Commands can be given by a variety of modalities in this kind of interface, including screens, voice, and manual marking of target items with a laser pointer. In this instance, the issue is that people with disabilities would find it challenging to reliably operate the robot through traditional modes. Brain signals have been suggested as a feedback modality in robotics systems for command and control in order to overcome this problem. While procedures based on mental images do not always involve the presentation of stimuli, these approaches usually use screens. When it comes to tasks requiring a high level of precision or for users with physical impairments, BCIs are especially helpful.

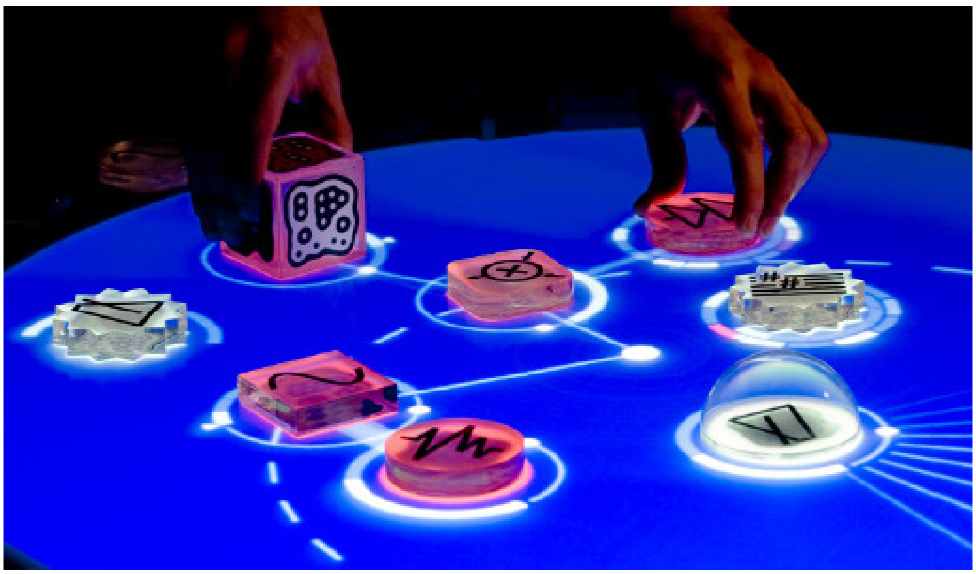

Figure 4: Tangible User Interface on a projected table.

The demonstration process in PbD can be greatly impacted by the selection of input modalities and, consequently, by the choice of distinct interfaces. For jobs involving physical manipulation or dexterity, for instance, utilizing a tangible interface might be more effective; on the other hand, tasks involving complicated spatial thinking or visualisation might benefit more from using an augmented reality interface. Similar to this, the sort of interface that is used can have a big influence on how well the user and the robot communicate the intended task. This can depend on a number of things, including the user's level of interaction requirements, familiarity with the interface, and technical proficiency. Of course, creating an interface that will allow for more versatile and adaptable robot programming is still the largest obstacle to be overcome.

Observational learning, kinesthetic teaching, and teleoperation settings are some of the most popular learning modalities utilized in PbD that are directly tied to human social interactive notions of teaching and robot demonstration.

1. As the name suggests, observational learning is a sort of learning where students gain knowledge by seeing a demonstration carry out a task. In that scenario, the primary input modalities are found in the fields of motion recording devices such as accelerometers and gyroscopes, as well as vision systems.

Humanoid robot applications can benefit greatly from the ability to replicate a user's movements on a robot through the use of three-dimensional motion capture technologies. Three primary parts make up the system: measuring human motion, translating human motion to a robot, and controlling robot motion. Based on how the observations are conducted, two forms of learning are distinguished: aided and direct. While the learner in the second scenario is given additional tools, such as Deep Neural Networks for optimisation to discover and track the observed interactions, the learner in the first scenario watches the video modality on their own. Sensing modalities including trackers, visual detectors, motion capture, and additional visual inputs are used in the assisted approaches.

2. Kinesthetic teaching is a mode in which the user uses haptic feedback and physical contact to direct the robot along a predetermined course. The robot is configured for easy control in the gravity correction mode in this modality. Physical contact introduces a more organic method of disseminating the knowledge required to demonstrate, replicate, or hone a skill. This allows the user to interact more tactilely and broadens his or her understanding of the robot's capabilities. As a means of providing demonstration, kinesthetic education offers a viable answer to the correspondence problems that pertain to various mapping obstacles, such as the function of mapping humans to robots and the ability to adapt to various settings. One of the disadvantages is that it can be difficult to teach activities through demonstrations that need for good coordination across several limbs.

An effective source for primitive generation can also be produced through kinesthetic instruction. It is also feasible to take advantage of the robot's ability to learn a task incrementally when it is in kinesthetic mode. Using a variety of learning modalities, incremental learning approaches offer a natural way to teach natural movements and guarantee the synchronization of intricate whole-body motions. Usually, learning starts with observation learning, which is teaching a robot how to move their entire body from a human demonstration. Next, the procedure entails honing kinesthetic movements.

3. Immersion teleoperation, which makes use of haptic interfaces and makes use of the robot's effectors and senses. Although there are limitations for the user, one benefit is that they can receive instruction remotely. Once more, the communication problem solution is the other benefit. The goal of the teleoperation learning modality is to limit the user's perception to that of the robot, in contrast to kinesthetic teaching, which restricts the user to the physical body of the robot. Joysticks and other remote control devices—including haptic devices—can be used for teleoperation. Teleoperation is primarily employed to convey the kinematics of motion in most circumstances, and its effectiveness is strongly connected with the teacher's capacity to utilize the remote control device, which naturally calls for training. Using AR and VR to teleoperation scenarios is another more modern way.

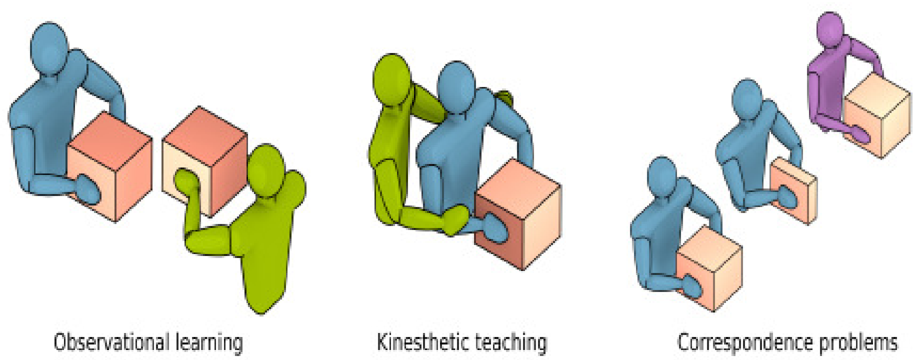

Figure 5: Input modalities and the correspondence challenges in robot Programming by Demonstration.

Each of the teaching interfaces and learning modalities previously discussed performs better than the others in some situations, making them appropriate for use in particular activities, although they all have drawbacks. The potential of combining these interfaces to benefit from the complementing information offered by each medium independently is being investigated.

4.0 Intuitive Robot Programming

The concept of Programming by Demonstration and the endeavor to create interfaces that require no prior knowledge or high expertise are the main points of reference when discussing Intuitive Robot Programming (IRP). This approach makes interfaces accessible to a wider range of users, including those without specialized programming knowledge or relevant experience. A primary obstacle in conventional robot programming is the intricacy of programming languages and the requirement for detailed, exact instructions. IRP seeks to make this process easier to understand and more approachable by enabling users to program robots using more organic methods.

4.1 The Teacher’s Role in Programming by Demonstration

As was mentioned in earlier sections, ensuring that the robot can acquire skills effectively by adjusting to uncertainty while operating and lowering the instructor's instruction load is essential to making LfD practicable in real-world applications. Consequently, learning to execute a demonstration alone is insufficient for the robot to understand how to perform better; generalization is an essential component of the learning process.

A more comprehensive approach to improving the entire robot programming concept should involve a detailed examination of the teacher's perspective and comprehension of the system. In contrast, Programming by Demonstration was approached from the standpoint of the robot and how it learns. One noteworthy argument in that regard, first put forth by Sena et al., is that the difference between the teacher's perception of the system's knowledge and its actual capabilities—caused by the previously mentioned low quality of the input data—remains largely unexplored and has the potential to be problematic, potentially leading to the following major issues:

- undemonstrated states
- ambiguous demonstrations
- unsuccessful demonstrations.

These problems may negatively impact the robot's integration and task familiarization during demonstration, and consequently, the repetition of skill behavior. To mitigate this effect, it is crucial that the users share certain fundamental mental components. This would allow the instructor to act in a way that would facilitate understanding of the issues surrounding the robot's learning process during the whole teaching phase.

4.2 Task-Level Learning and Collaborative Programming

Utilizing high-level behavioral techniques to convey knowledge and skills for a task through method addition and combination is another approach to creating intuitive interfaces for robot programming. Task-level techniques and visual programming languages are being used by several recent researchers to address the skill handling problem. One such approach combines PbD with Task Level Programming (TLP). The robot is designed to carry out a task using TLP in a way that is comparable to how a human would carry out the same work. Rather than giving specific motor orders, the programmer describes a set of abstract activities that the robot needs to execute in order to finish the task. Natural language, graphical user interfaces, or other high-level programming languages can be used to define these abstract behaviors. The TLP's benefits increase PbD by:

- intuitive with robust results, since the error-handling can be incorporated into individual skills

• safe, since the skills can be certified,

• generic, the skills can be parameterised for different task variations.

As per Steinmetz et al., the advantages of both approaches are the only things left over when combining the TLP and PbD into Task-Level Programming by Demonstration (TLPbD). Through the use of Planning Domain Definition Language (PDDL) descriptions of the available abilities, TLPbD allows professionals to semantically annotate robot skills with their conditions and effects, enabling non-experts to realize online recognition from demonstrations. This is made possible by the semantic skill recognizer, the system's central component, which provides parameterized skills to the TLP interface in response to the latter's signal confirming that the system's current state is accurate. The world model, which is updated by the world observer and includes more abstract representations than just raw trajectories, is the basis on which the recognizer builds the skills. It also includes the PDDL descriptions of the various skills.

Collaborative Incremental Programming (CIP), a framework developed by Willibald et al., allows the user and robot to program difficult tasks together in an attempt to create a more participatory robot programming interface.

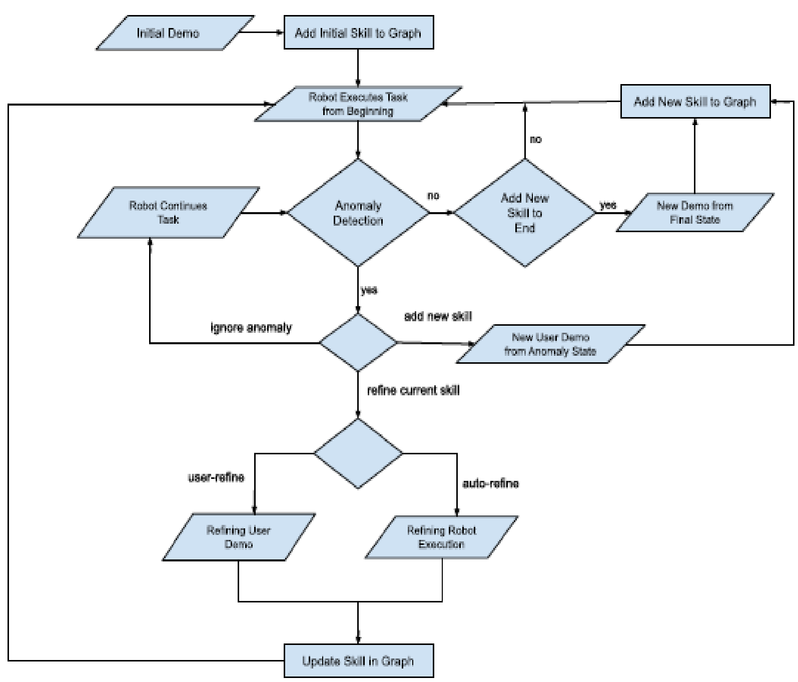

Figure 6: Collaborative programming with on-line anomaly detection for interactive teaching where the user can incrementally program the robot by adding new skills or refining existing ones.

The capacity to exhibit new skills or improve upon already acquired ones, as well as an online anomaly detection system, are two of the key features made possible by this framework. In order to improve the user's understanding regarding the system, the entire process is visualized using a high-level graph that depicts every state and sequence of activities related to the current task.

Any local perturbation is balanced out using low-level statistical skill encoding, and departures from the desired behavior can be detected via the model's intrinsic probability distribution. Figure 6's flow chart illustrates the algorithm's workflow.

The following are the CIP framework's primary functions:

1. Probabilistic encoding of demonstrations, allowing the system dynamics to be integrated without requiring human intervention. To do this, the user records a second sequence, has the robot replicate the user's demonstration trajectory, and uses sensor readings to learn variability through multiple demonstrations and robot replication. Higher deviations are accepted during the execution of parts with significant variability, although accuracy is required for parts with less variability. The process is strengthened by this tactic.
2. Online anomaly detection, which uses a real-time monitoring segment to continually compare the commanded and measured sensor modalities. In the event that the latter is found, it is categorized into one of the following sensor categories: grasp status, gripper opening, force/torque, end effector position/orientation, and so on. This feature enables the anomaly detection algorithm to leverage the robot's proprioceptive sensory systems and establish a bidirectional dependency between the robot and the learning model.
3. Incremental graph creation provides a more interactive method of adjusting the task-graph to any potential changes by letting the user choose between two actions: either adding a new skill to the graph or improving an existing skill.
4. The teaching process is integrated with the application of learned behavior within a cooperative framework. At crucial task skill transition times, the choice states are automatically inserted. This feature removes perceptual aliasing and reduces the amount of computer time required for the decision-making process. As such, the chance of choosing the wrong talent is reduced.

5.0 Immersive Computing Technologies and Augmented Reality

The goal of the fast developing field of immersive computing technologies, which includes a wide range of Extended experience (XR) technical paradigms, is to create digital worlds that not only replicate but also improve our physical experience. Immersion computing's primary goal is to give customers an entirely immersive and interactive experience that appeals to their senses—sight, hearing, touch, and even smell.

Any technology that generates a virtual experience that immerses the user, either fully or partially, in a digital world falling within the Virtuality Continuum is referred to by this term. This can range from Augmented Reality (AR) apps that superimpose digital content on the physical world to Virtual Reality (VR) headsets that produce completely immersive 3D settings. To do it, a variety of tools and methods must be used, including sophisticated graphics processing, 3D modeling, and sensors for motion tracking, haptic feedback, and even odor producers.

As was already established, the category of Immersive Computing Technologies comprises a variety of techniques, each with a distinct focus and technology stack. The following are the most popular subcategories of immersive computing:

• Virtual Reality (VR): Virtual reality stands to be the most popular subset of Immersive Computing. It produces a digital environment that is totally immersive and isolated from reality. Wearing a VR headset that tracks the user's head movements and generates a 3D world in real-time is a common way to achieve this.

• Augmented Reality (AR): Using computer visuals and virtual objects, AR superimposes digital content on the real world to enhance or augment the user's perception of their surroundings. Usually, a mobile device's camera is used to accomplish this, tracking and identifying real-world objects and superimposing digital content on top of them.

• Mixed reality (MR): MR includes both augmented reality (AR) and augmented virtuality (AV), and it spans the whole range between the fully virtual and the actual world. Digital content is superimposed onto the physical world in mixed reality (MR) to create an immersive experience and facilitate more lifelike interaction concepts. It blends digital content with the physical environment in a more interactive and context-aware way than just overlaying objects, like AR does. A more immersive and smooth transition between the virtual and physical worlds is possible in mixed reality (MR) because digital material is aware of and able to interact with the real environment.

The framework of VR is closely connected with the term of immersion as well, since in VR is defined as: "real-time interactive graphics with 3D models, combined with a display technology that gives the user the immersion in the model world and direct manipulation". Furthermore, the VR applications can be categorized into 3 different types depending on the level of immersion they introduce: nonimmersive, immersive and semi-immersive. It is stated that an AR system should combine real and virtual objects in a real environment, operate in real-time in an interactive way, maps and registers (aligns) real and virtual objects with each other. Regarding the definition content for MR, it was given initially in but with the marketing policies of today’s big companies the term can be confused with a slightly diverged one.

Immersion computing is used to create content using a variety of newly developed technologies. These technologies' primary feature is a type of display that allows the user to interact with virtual features that are projected onto the screen to expand reality. These are a few of those display frameworks:

- To create XR interfaces, Head-Mounted Displays (HMDs) are the most popular method. HMDs are made up of a series of sensors that track the user's head motions in real time, together with a visual display. The visual display usually renders a 3D or 2D virtual environment in real-time using stereoscopic or monoscopic pictures. The tracking sensors detect head movements of the user and modify the virtual world based on those motions using a variety of technologies, including inertial measurement units (IMUs), optical tracking systems, and magnetic tracking systems. The HTC Vive Pro 2, Oculus Quest 2, Microsoft's HoloLens 2, and other cutting-edge devices are included in that group.
- The Head-up Display (HUD), a transparent display fitted on the windshield, helmet visor, or eyeglasses of a car, allows information to be seen without diverting the user's attention from their point of view. HUDs were first used in military aircraft to give pilots vital flying information without requiring them to glance down at the instrument panel, such as airspeed, heading, and altitude. HUDs are employed in many different industries these days, such as gaming, aviation, and automobiles. Automotive behemoths like BMW, Mercedes, Tesla, and others are heavily investing in research and development of those kinds of displays, which is leading to an increasing number of applications in the market. HUDs are commonly employed in aviation since this type of technology has numerous military uses, including targeting systems, data display, semi-autonomous pilot guidance systems, and more.
- Room-Scale Immersion: This technique uses cameras and sensors to follow a user's movements in a real-world setting, enabling more natural movement and interaction with the digital world. To offer a more immersive experience, VR headsets are frequently combined with this technology. The Oculus Rift S, a mid-range VR headset

with room mapping capabilities up to 400 square feet of playable space, is one example of gear.

- Mobile Devices, such as smartphones and tablets are increasingly being used to create immersive experiences.

5.1 The Reality-Virtuality Continuum and Extended Reality (XR)

The collective sensory and perceptual experiences that humans gather from and interpret from the environment around them can be viewed as reality. These experiences are filtered by our senses of sight, hearing, touch, taste, and smell and are molded by our biological and cognitive capacities as well as our cultural and social settings. Furthermore, the culmination of advances in various scientific fields (computer science, electronics, neuroscience, psychology, etc.) has resulted in immersive extensions of reality, which represent the Extended Reality (XR) paradigms, including virtual reality (VR) and augmented reality (AR).

Establishing precise limits between the many categories contained in the general category of Immersive Computing Technologies and XR is crucial because the phrases Virtual Reality (VR) and Augmented Reality (AR) are frequently misconstrued and misused. To that end, Milgram et al.'s work represents a significant attempt to pinpoint a spectrum that will serve as a border between several realities.

The ability of AR systems to leverage the real environment is a major advantage over VR counterparts. First, Milgram et al. provided an official definition of the Mixed Reality framework, attempted to categorize the many forms of Visual Displays, and identified the specific examples. A particular class of Virtual Reality (VR) technologies are Mixed Reality (MR) visual displays, which blend the actual and virtual worlds together along the "virtuality continuum." From fully virtual to fully actual worlds, there is a continuum. Arguably the most widely used Mixed Reality (MR) technology, Augmented Reality (AR) modifies the display of an actual world by adding virtual computer visuals. Augmented Virtuality (AV) is the term used to describe the other situation on the virtuality continuum.

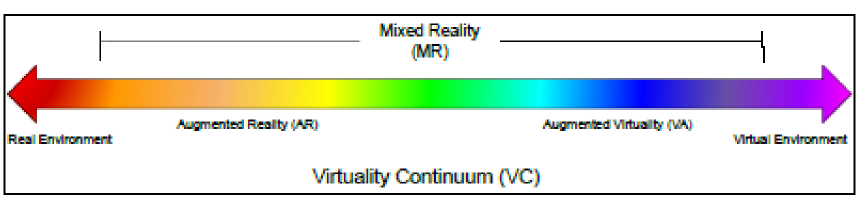

Figure 8: The Virtuality Continuum

The Viruatlity Continuum hyperspace has three main attributes that are defined in

• Reality: there are the computer generated virtual environments that were created artificially and the primarily real ones.

• Immersion: the need for the observer to be completely immersed within the interacted environment should not determine the ability to display both virtual and real environments.

• Directness: this concerns the ability to represent the world objects, if they are viewed directly or by using an electronic image synthesis process.

The Milgram et. al are trying to classify the different MR displays into 6 different categories. These classes are the following:

1. Non-immersive monitor-based AR displays, upon which Computer Graphic (CG) images are overlaid or in more modern displays holographic units.
2. Similar to previous using video displays but this time using Immersive HMDbased AR displays.
3. HMD-based AR systems, incorporating Optical See-Through (OST-HMD) and video see-through (VST-HMD).
4. Monitor-based Augmented Virtuality (AV) systems, with CG world substratum, employing superimposed video reality.
5. Completely graphical immersive or partially immersive systems, employing superimposed video or texture mapped reality.
6. Partially immersive AV systems with completely graphical like large screen displays, enabling as well real-object interactions.

5.2 Augmented Reality Approach in Robotics

In addition to knowing how to handle potential use cases, designers of augmented reality interfaces need also be aware of the full range of advantages that this technology may provide when correctly incorporated into a particular framework. Having said that, in order to fully utilize the aforementioned advantages in every application, the appropriate strategy should carefully consider the methodology environment that needs to be put into place and utilize the appropriate input modalities. In light of this, a thorough analysis of the taxonomy established in offers a thorough framework of the data required as a foundation for formulating a plan for creating an AR interface for HRI.

Table 1 shows the eight distinct dimensions that make up this taxonomy. A few of the difficulties raised in the paper might serve as models for further research, including the need to lessen the cognitive load on an interface, the trade-offs associated with augmented reality safety, certain hardware and technological constraints, bridging the gap between studies and systems, and novel approaches that could be applied in that situation.

Table 1: AR interfaces and they use in robotics

01	Approaches of AR for HRI	• Location of the augmentation device (on-body, on-environment, on-robot) • Target location of visual augmentation (augmented surroundings and augmented robot)
02	Characteristics of augmented robots	• Form factor/ type of robot (robotic arms, humanoid, drones, mobile robots etc.) • Relationship with regard to the number of users and robots incorporated in the interface • Scale of the robot (small, tabletop, body-scale, large) • Proximity to objects (near, co-location from distance, semi-remote, far)
03	Purpose and benefits	• Programming and control (facilitate programming, real-time control) • Understanding, interpretation, communication (improve safety, increase expressiveness, communicate intent)
04	Types of information	• Internal information • External information • Plan and activity • Supplemental content
05	Design elements and strategies of the interface	• User interface and widgets (menus, information panels, annotations and labeling, controls and handles, displays and monitors) • Spatial references and visualisation (points and location, path and trajectories, areas and bounds etc.) • Embedded visual effects (Anthropomorphic effects, virtual replicas and ghost effects texture mapping effects)
06	Interaction modalities	• Tangible input • Touch input • Controllers • Proximity by implementing trackers • Gaze input • Voice commands • Gesture input
07	Applications of AR interfaces in Robot Programming	• Domestic and everyday ese • Design and creativity tasks • Medical and health • Remote collaboration • Education and training • Mobility and Transportation • Industry • Social interaction • Data physicalisation • Entertainment • Search and Rescue
08	Evaluation Techniques	• Demonstration • Technical evaluation • User studies

A detailed overview of the latest developments in robotic Augmented Reality. Human-robot interaction (HRI), motion planning and control, medical robotics, and multi-agent systems are the three main robotics application domains that are covered in this survey of AR applications in robotics.

The application of AR to help three patients with different liver diseases during liver resections. Virtual Surgical Planning (VSP) and VR-Render software were used to create a three-dimensional (3D) virtual representation of each patient's abdominal cavity. The liver resection and port placement were then guided by the model during the procedure. Additionally, laparoscopic ultrasound was utilized to determine the margins of resection and to assess the tumor's location. The outcomes demonstrated that accurate tumor dissection with safe and accurate identification of the main vascular systems was made possible by AR-guided liver resection.

The use of AR in networks of interconnected physical systems and human-machine communication has been accelerated by the rise of Industry 4.0. The integration of augmented reality interfaces in the field of industrial robotics is a subject of growing scientific interest, with several notable examples. AR has several applications in the industrial sector, including robot programming, robot teleoperation, workspace inspection, maintenance, employee training, and more.

The AR interface telematic control system allows for remote work environment inspection and maintenance. The telematic user interface is made up of two windows: one that shows various perspectives of the workspace, and another that displays control elements for data presentation, customisation, work process analysis, and remote robot control in addition to system status information. As a result, teleoperators can use augmented reality modalities to remotely control the robot and obtain situational awareness. The interface also makes it possible to display data with AR in an intuitive way. Data with positional and value information can be represented using lines, boxplots, or directional symbols.

The system explains the creation of a virtual robot system that uses HoloLens to enable human-robot interaction using gestures and virtual tools. KRAgius and iiwa, two virtual robot models, were used; their geometric models faithfully captured real-world parameters. Application Manager, Trajectory Planner, Geometrical Path Planner, and Simulations make up the system. It enables the robot to be programmed for a range of activities while accounting for its kinematic model and constraints. Gestures and virtual items, such menus and spatial maps, are used to interface with robots. The system is also capable of managing path planning and modification, including minimum-time trajectory planning and smoothing. Trajectory planning and robot model replacement are made simple by the application (Figure 9 and Figure 10).

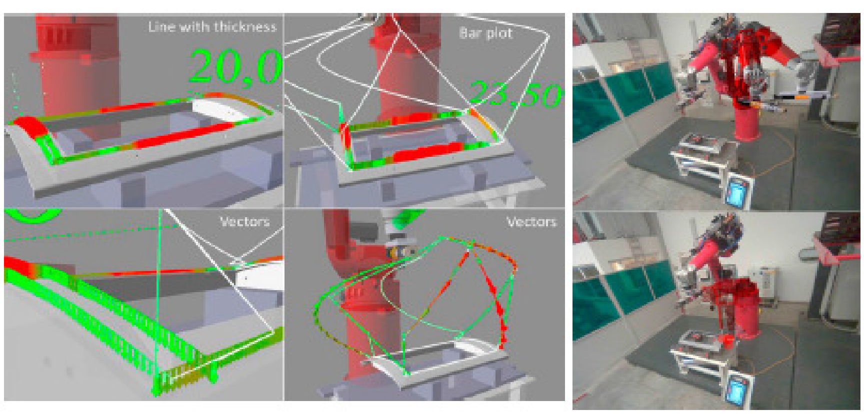

Figure 9: Spatial data visualisation (right) and pre-visualised movement with overlay with possible collision

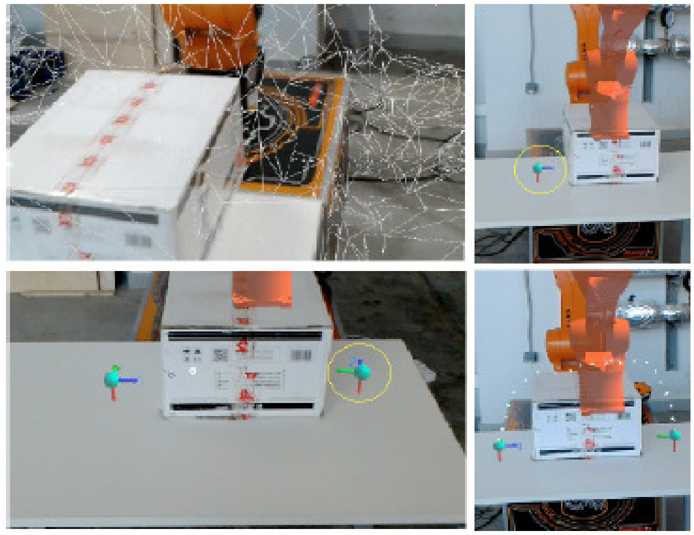

Figure 10: User path represented with pointer and point to point with collision avoidance, scanning, goal, points setting and path planning

6.0 Summary of the Proposed Approach

Our primary review goal is to develop a framework that will allow for the demonstration of an Augmented Reality interface that facilitates simple robot programming. Robot programming includes immersive features through the use of an augmented reality interface, which may improve the user's intuition about the robot's capabilities and how to effectively carry out manipulation and programming activities. Moreover, the architecture for modular implementation presents the theoretical notion of five distinct capabilities.

It would be possible to incorporate new features into the system, such as an integrated task graph functionality. It might be used in conjunction with a carefully planned user study to demonstrate the advantages of utilizing an augmented reality interface for a particular use-case. The components might be arranged by the user in a more flexible or preset way, and the virtual routes of the objects could be rendered using the AR interface. Additionally, the user could verify where the holographic items were placed in the scene and reposition them in a self-arranged layout.

References

1. Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In Proceedings of the Twenty-First International Conference on Machine Learning, ICML '04, page 1, New York, NY, USA, 2004. Association for Computing Machinery. doi:10.1145/ 1015330.1015430.

2.Baris Akgun and Kaushik Subramanian. Robot learning from demonstration : Kinesthetic teaching vs . teleoperation. 2011.

3. Darrin Bentivegna, Christopher Atkeson, and Gordon Cheng. Learning tasks from observation and practice. Robotics and Autonomous Systems, 47:163–169, 06 2004. doi:10.1016/j.robot.2004.03.010.

4. Aude Billard, Sylvain Calinon, R¨udiger Dillmann, and Stefan Schaal. Robot Programming by Demonstration, pages 1371–1394. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi:10.1007/ 978-3-540-30301-5_60.

5. Gary Bishop and Henry Fuchs. Research directions in virtual environments: Report of an nsf invitational workshop, march 23-24, 1992, university of north carolina at chapel hill. SIGGRAPH Comput. Graph., 26(3):153ˆa177, aug 1992. doi:10.1145/142413.142416.

6. A. Billard and D. Grollman. Robot learning by demonstration. Scholarpedia, 8(12):3824, 2013. revision #138061. doi:10.4249/ scholarpedia.3824.

7. Istv´an Barakonyi and Dieter Schmalstieg. Exploiting the physical world as user interface in augmented reality applications. 03 2023.

8. Sebastian Blankemeyer, Rolf Wiemann, Lukas Posniak, Christoph Pregizer, and Annika Raatz. Intuitive robot programming using augmented reality. Procedia CIRP, 76:155–160, 2018. 7th CIRP Conference on Assembly Technologies and Systems (CATS 2018). URL: https://www.sciencedirect.com/science/article/pii/S2212827118300933.

9. S. Calinon. Robot Programming by Demonstration: A Probabilistic Approach. Engineering sciences. CRC, 2009. URL: https://books. google.de/books?id=7l65QwAACAAJ.

10. Sylvain CALinon. A tutorial on task-parameterized movement learning and retrieval. Intelligent Service Robotics, 9, 01 2016. doi:10.1007/s11370-015-0187-9.

11. Sylvain Calinon. Learning from Demonstration (Programming by Demonstration), pages 1–8. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018. doi:10.1007/978-3-642-41610-1_27-1.

12. Pietro Cipresso, Irene Alice Chicchi Giglioli, Mariano Alca˜niz Raya, and Giuseppe Riva. The past, present, and future of virtual and augmented reality research: A network and cluster analysis of the literature. Frontiers in Psychology, 9, 2018. URL: https://www.frontiersin. org/articles/10.3389/fpsyg.2018.02086, doi:10.3389/ fpsyg.2018.02086.

13. Sylvain Calinon and Dongheui Lee. Learning Control, pages 1261– 1312. Springer Netherlands, Dordrecht, 2019. doi:10.1007/ 978-94-007-6046-2_68.

14. Matt Coppinger. Unlocking workforce productivity with spatial computing. (accessed 12-03-2023). URL: https://octo.vmware.com/ unlocking-workforce-productivity-spatial-computing/.

15. Maya Cakmak and Andrea L. Thomaz. Eliciting good teaching from humans for machine learners. Artificial Intelligence, 217:198–215, 2014.

16. Sonia Chernova and Manuela Veloso. Multi-thresholded approach to demonstration selection for interactive robot learning. In Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction, HRI

’08, page 225ˆa232, New York, NY, USA, 2008. Association for Computing Machinery. doi:10.1145/1349822.1349852.

17. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological), 39(1):1– 22, 1977.

18. Staffan Ekvall and Danica Kragic. Robot learning from demonstration: A task-level planning approach. International Journal of Advanced Robotic Systems, 5(3):33, 2008. arXiv:https://doi.org/ 10.5772/5611, doi:10.5772/5611.

19. Hind Gacem, Gilles Bailly, James Eagan, and Eric Lecolinet. Finding objectsfaster in dense environments using a projection augmented robotic arm. In Human-Computer Interaction INTERACT 2015, pages 221–238, Berlin, Heidelberg, 2015. doi:10.1007/978-3-319-22698-9_15.

20. Florent Guenter, Micha Hersch, Sylvain Calinon, and Aude Billard. Reinforcement learning for imitating constrained reaching movements. Advanced Robotics, 21:1521 – 1544, 2007.

21. Dominic Gorecky, Mathias Schmitt, Matthias Loskyll, and Detlef Z¨uhlke. Human-machine-interaction in the industry 4.0 era. In 2014 12th IEEE International Conference on Industrial Informatics (INDIN), pages 289–294, 2014. doi:10.1109/INDIN.2014.6945523.

22.Thomas Hellstr¨om and Suna Bensch. Understandable robots - what, why, and how. Paladyn, Journal of Behavioral Robotics, 9(1):110–123, 2018. URL: https://doi.org/10.1515/pjbr-2018-0009 [cited 2023-03-10], doi:doi:10.1515/pjbr-2018-0009.

23. Jakob H¨orbst and Horst Orsolits. Mixed reality hmi for collaborative robots. In Roberto Moreno-D´ıaz, Franz Pichler, and Alexis Quesada- Arencibia, editors, Computer Aided Systems Theory – EUROCAST 2022, pages 539–546, Cham, 2022. Springer Nature Switzerland.

24. Thomas Howard, Stefanie Tellex, and Nicholas Roy. A natural language planner interface for mobile manipulators. pages 6652–6659, 05 2014. doi:10.1109/ICRA.2014.6907841.

25. A.J. Ijspeert, Jun Nakanishi, and Stefan Schaal. Learning attractor landscapes for learning motor primitives. volume 15, pages 1523–1530, 01 2002.

26. Nikolay Jetchev and Marc Toussaint. Fast motion planning from experience: Trajectory prediction for speeding up movement generation. Autonomous Robots, 34, 01 2013. doi:10.1007/ s10514-012-9315-y.

27. Petar Kormushev, Sylvain Calinon, and Darwin G. Caldwell. Robot motor skill coordination with em-based reinforcement learning. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3232–3237, 2010. doi:10.1109/IROS.2010.5649089.

28. Jens Kober and Jan Peters. Policy search for motor primitives in robotics. Mach. Learn. J, 84:171–203, 01 2008. doi:10.1007/ s10994-010-5223-6.

29. Daniel Kappler, Peter Pastor, Mrinal Kalakrishnan, Manuel Wuthrich, and Stefan Schaal. Data-driven online decision making for autonomous manipulation. In Proceedings of Robotics: Science and Systems, Rome, Italy, 2015.

30. Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In S. Solla, T. Leen, and K. M¨uller, editors, Advances in Neural Information Processing Systems, volume 12. MIT Press, 1999. URL: https://proceedings.neurips.cc/paper/1999/file/ 6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

31. Henrich Kolkhorst, Joseline Veit, Wolfram Burgard, and Michael Tangermann. A robust screen-free brain-computer interface for robotic object selection. Frontiers in Robotics and AI, 7, 2020. URL: https://www.frontiersin.org/articles/10. 3389/frobt.2020.00038, doi:10.3389/frobt.2020.00038.

32. YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation, 2018. arXiv:1707.03374.

33. Florian Leutert and Klaus Schilling. Augmented reality for telemaintenance and -inspection in force-sensitive industrial robot applications. IFAC-PapersOnLine, 48(10):153–158, 2015. 2nd IFAC Conference on Embedded Systems, Computer Intelligence and Telematics CESCIT 2015. URL: https://www.sciencedirect.com/ science/article/pii/S240589631500991X, doi:https:// doi.org/10.1016/j.ifacol.2015.08.124.

34. Hangxin Liu, Yaofang Zhang, Wenwen Si, Xu Xie, Yixin Zhu, and Song-Chun Zhu. Interactive robot knowledge patching using augmented reality. pages 1947–1954. IEEE Press, 2018. doi:10.1109/ICRA.2018. 8462837.

35. Bernard Michini and Jonathan P. How. Bayesian nonparametric inverse reinforcement learning. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, Machine Learning and Knowledge Discovery in Databases, pages 148–163, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

36. Carl Rasmussen. The infinite gaussian mixture model. In S. Solla, T. Leen, and K. M¨uller, editors, Advances in Neural Information Processing Systems, volume 12. MIT Press, 1999. URL: https://proceedings.neurips.cc/paper/1999/file/ 97d98119037c5b8a9663cb21fb8ebf47-Paper.pdf.

37. Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pages 627–635, 2011.

38. Harish Ravichandar, Athanasios S. Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. Annual Review of Control, Robotics, and Autonomous Systems, 3(1):297–330, 2020. doi:10.1146/ annurev-control-100819-063206.

39. Yasaman S. Sefidgar, Prerna Agarwal, and Maya Cakmak. Situated tangible robot programming. In Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, HRI '17, pages 473–482, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/2909824.3020240.

40. Eric L. Sauser, Brenna D. Argall, Giorgio Metta, and Aude G. Billard. Iterative learning of grasp adaptation through human corrections. Robotics and Autonomous Systems, 60(1):55–71, 2012.