Memory-centered and Affordance-based Framework for Mobile Manipulation

Christoph Pohl*, Fabian Reister*, Fabian Peller-Konrad and Tamim Asfour

***Abstract*— Performing versatile mobile manipulation actions in human-centered environments requires highly sophisticated software frameworks that are flexible enough to handle special use cases, yet general enough to be applicable across different robotic systems, tasks, and environments. This paper presents a comprehensive memory-centered, affordance-based, and modular uni- and multi-manual grasping and mobile manipulation framework, applicable to complex robot systems with a high number of degrees of freedom such as humanoid robots. By representing mobile manipulation actions through affordances, i. e., interaction possibilities of the robot with its environment, we unify the autonomous manipulation process for known and unknown objects in arbitrary environments. Our framework is integrated and embedded into the memory-centric cognitive architecture of the ARMAR humanoid robot family. This way, robots can not only interact with the physical world but also use common knowledge about objects, and learn and adapt manipulation strategies. We demonstrate the applicability of the framework in real-world experiments, including grasping known and unknown objects, object placing, and semi-autonomous bimanual grasping of objects on two different humanoid robot platforms.**

I. Introduction

Due to their targeted application in dynamic and human-centered environments, assistive humanoid robots need to be able to robustly handle everyday situations like clearing cluttered kitchen tables or setting the dinner table. This is especially challenging considering the unknown objects and unstructured surroundings these tasks involve. The integration of mobility and manipulation capabilities is essential to enhance the versatility and adaptability of robots in such real-world scenarios.

Employing the concept of affordances [1], i. e., interaction possibilities of an agent with its environment, from cognitive psychology to the scenarios is advantageous, especially in the case of unknown objects. It allows for the representation of potential action possibilities without any prior knowledge of the objects. By assigning action possibilities as properties to relevant objects and locations, affordances provide a way to reason about the environment in terms of what can be done with objects, rather than what the objects are. Therefore, affordances allow for a more flexible and adaptable approach to interacting with the environment, as they enable robots to identify and utilize potential action possibilities without relying on specific object features or properties. To improve the performance of manipulation actions, it is advantageous to have access to "common knowledge", i. e., knowing which objects can be grasped, where objects have been seen the last time, or where they can be usually found and what is the most successful way to grasp objects that the robot already knows. Additionally, having a recollection of past action executions, their parametrization, and their outcomes promotes explainability and facilitates learning from experience.

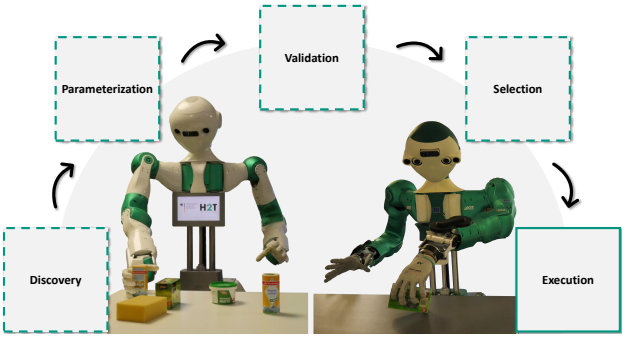

Fig. 1: The humanoid robots ARMAR-DE and ARMAR-6 grasping and placing objects using our framework.

However, the amount of expert knowledge required to implement robotic applications in these unpredictable, dynamic, and complex settings slows the progress of the deployment of robots into daily environments. On the other hand, most implementations and systems are highly specialized to specific scenarios and contexts. Therefore, there is a need for a framework that facilitates the flexible design and implementation of mobile manipulation tasks involving known and unknown objects in unstructured environments and that can fully leverage the advantages of a cognitive memory architecture.

We propose a memory-centered, affordance-based mobile manipulation framework that unifies the task description of various manipulation actions (e. g., pick-and-place tasks) across different situations and robotic platforms. It allows for the autonomous and semi-autonomous generation and execution of uni- and multi-manual manipulation actions while being flexible enough to support customization of the single steps to the user's needs and various scenarios. The coupling of our framework with a memory-centric cognitive architecture [2] enables introspection and state disclosure, as well as learning from experience. We provide several use cases of our approach and show their application to

* The authors contributed equally to this paper.

The research leading to these results has received funding from the German Federal Ministry of Education and Research (BMBF) under the competence center ROBDEKON II (13N16539), the Carl Zeiss Foundation through the JuBot project and the Baden-Württemberg Ministry of Science, Research and the Arts (MWK) as part of the state's "digital@bw" digitization strategy in the context of the Real-World Lab "Robotics AI".

The authors are with the High Performance Humanoid Technologies Lab, Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany{pohl, asfour}@kit.edu

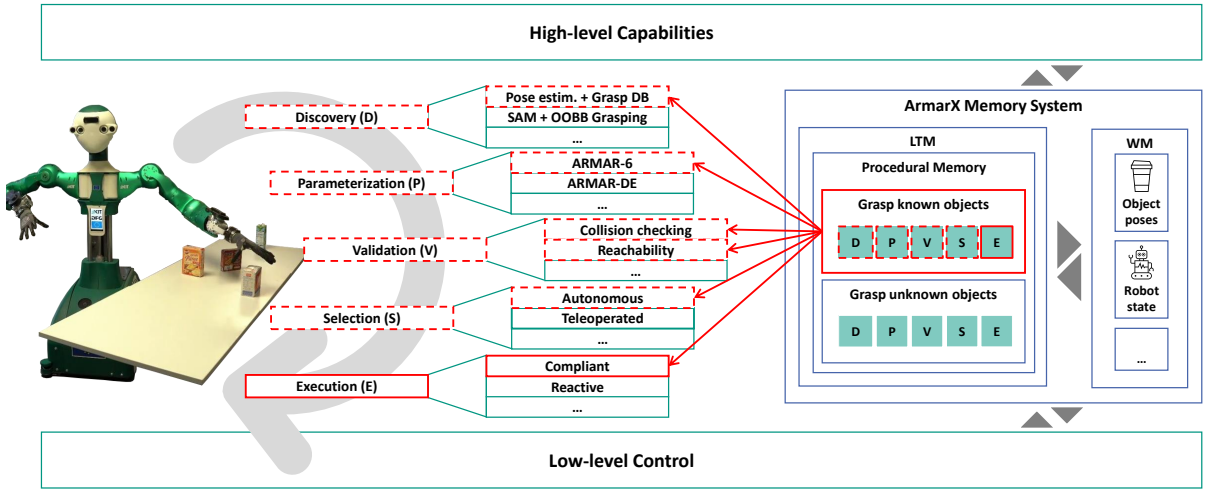

Fig. 2: Embedding of our framework into the memory-centric cognitive architecture [2]. Several strategies that implement the five steps of the mobile manipulation architecture are connected to the procedural memory. We assume the robot to be in the *Execution* step. Already performed phases of the high-level skill "Grasp known object" are marked with dotted borders.

the uni- and bimanual grasping and placing of known and unknown objects on the humanoid robots ARMAR-6 [3] and ARMAR-DE.

II. Related Work

There exist multiple frameworks that aim to simplify the creation and description of robotic tasks in a unified manner, like the *Statechart* [4] concept of ArmarX [5] or *SMACH* of ROS [6]. For example, in [7] a modular framework for programming robots based on tasks and skills, which is organized into abstraction levels, is introduced. It leverages a world model to enhance planning and is based on the hierarchical structure of robot capabilities. The authors of [8] propose a three-layered architecture, called *LAAIR*, for autonomous robots performing real-world tasks. It sequences modular skills using a deliberate control layer and offers reactive control options, providing flexibility in task execution. The *Affordance Templates Task Description Language* ([9], [10]) is particularly relevant to our approach, as it also employs affordances to describe manipulation tasks in a robot-agnostic manner. Similarly, we developed our own *task description* based on scene affordances and abstract end-effector poses. However, their approach is centered around semi-autonomous manipulation requiring the strong involvement of a human operator through *RViz*. Our method focuses on – but is not limited to – autonomous task execution through the combination of a task description with a memory-centered software architecture that can easily adapt and instantiate actions for different manipulation tasks and robots.

Even though these works offer valuable assistance in the design of general robotic applications, a considerable amount of work is necessary to adapt them to specific scenarios and different robot architectures. Therefore, multiple works consider integrated software architectures that can handle different tasks and environments. The authors of [11] present an integrated hardware and software system for mobile manipulation in industrial applications. It facilitates the use of pre-computed grasp candidates and a modular roadmap planner for task programming. In [12], the *Acromovi* framework is extended to enable distributed mobile manipulation by combining a manipulator with a mobile base. A modular, general-purpose software framework for mobile manipulation in household environments is introduced in [13]. It covers navigation, visual perception, manipulation, human-robot interaction, and high-level autonomy, demonstrating its versatility in various tasks and environments. In [14], a mobile manipulation system for domestic environments is presented. This system incorporates natural language processing, perception, navigation, and integrated motion and grasp planning, highlighting its success in the Robocup@Home competition. The *BART* framework, a behavior-based architecture for mobile manipulation tasks, is introduced in [15]. BART has a focus on ready-to-use software components for execution speed, quality, and performance.

Some other approaches specifically focus on reactive grasping and develop special control architectures for such applications. The authors of [16] present a reactive control scheme for vision-based mobile grasping of unknown objects. This approach tracks and extracts grasping regions for objects, enabling reach and grasping motions. The approach of [17] focuses on a control architecture for reactive pick-and-place tasks with a one-armed mobile manipulator. Their control scheme facilitates the execution of grasps while the mobile base of the robot is still on the move.

In contrast to the existing literature, our focus lies on the combination of the integrated, memory-centered software architecture with our affordance-based task description. This enables the modular adaption and extension of multi-handed manipulation tasks while being able to access the full advantages of the cognitive robot memory. Our aim was to develop a framework that facilitates autonomous mobile manipulation in unstructured environments.

III. Framework for Mobile Manipulation

Based on the Interpretable Data Format (*IDF*), which is necessary to integrate our framework to the memory-centric cognitive architecture [2] of ArmarX, we developed a task description for mobile manipulation tasks in unstructured environments for the autonomous execution of uni- and bimanual actions across different robotic platforms. An overview of our approach can be seen in Figure 2.

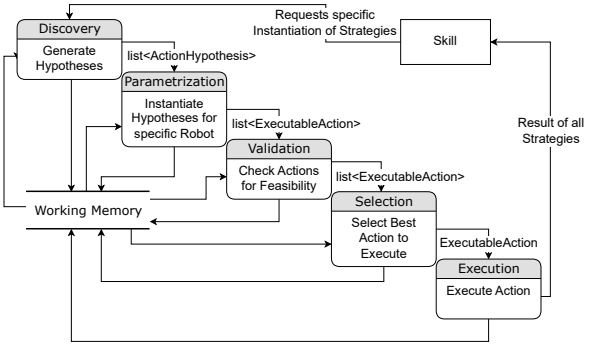

Fig. 3: Data flow in our framework visualizing the interaction of the *IDF* task description with the system architecture.

A. Design Principles

For the development and implementation of our software framework, we followed multiple design principles that were driven by the goals of our approach:

- ***memory-centered***: explainability, learning from experience
- ***affordance-based***: unification of different action types for known and unknown objects
- ***modular***: flexibility and extensibility
- ***robot-agnostic***: usability across different robots
- ***end-effector-based***: allows definition of unimanual and multi-manual actions

B. Task Description using IDF

The description of a manipulation task is – similar to [9] – based on the concept of affordances and is defined in terms of *IDF* objects, which can be seen in Figure 4. As the concept of affordances is, per definition, agent-specific, we define a robot-agnostic counterpart to an `Affordance` to be an `ActionHypothesis`. An action hypothesis is, therefore, an abstract end-effector pose connected to an `ActionType`, like *Grasp*, *Place*, or *Push*. This facilitates the extraction of action candidates from visual perception independent of the robotic system this action will be executed on.

The main object relevant to the execution is the `ExecutableAction`, which contains all relevant and necessary information for a specific robot. This object can contain up to n `Unimanual`, which consist of relevant information for a single end-effector. An affordance-based manipulation action is defined to be a `HandTrajectory` (i. e., a *Hand-Finger-Trajectory*) that is executed at the `executionPose` (i. e., an end-effector pose). Additionally, a pre- and retract pose can be defined, which will be approached before and after the execution of the action trajectory, respectively. After an `ExecutableAction` has been executed, its result and all relevant execution information will be saved in a `ExecutedAction` for storage and introspection in the memory.

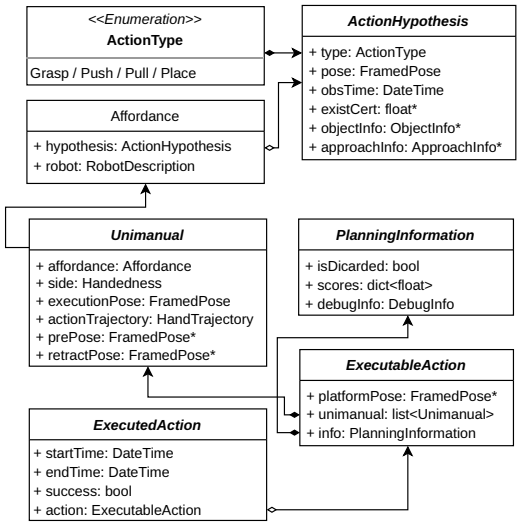

Fig. 4: Simplified class diagram of the *IDF* task description. Types marked with a "*" are optional.

C. System Architecture

Our integrated architecture is designed to facilitate the autonomous discovery and execution of mobile manipulation actions in unknown and unstructured environments while being flexible enough to adapt to different applications, situations, and robots. To this end, we split the overall task of generating and executing actions into five distinct steps:

- ***Discovery*** of affordances (`ActionHypothesis`) from e. g., visual perception (images and point clouds) or prior scene knowledge.
- ***Parameterization***. An `ExecutableAction` is derived from a `ActionHypothesis` for a specific robot, which contains all necessary information for execution (i. e., robot base poses, `HandTrajectory`, etc.).
- ***Validation*** of all `ActionHypothesis` by checking for feasibility e. g., reachability, correct handedness, approach direction, collision checking.
- ***Selection*** of the best `ExecutableAction` based on multiple criteria (e. g., execution height, proximity to previously executed actions, execution side, platform movement).
- ***Execution*** of the `ExecutableAction`. In this paper, we use an approach similar to [18] combined with the navigation of [19].

An overview of the data flow of our architecture can be seen in Figure 3. All steps are implemented using the *Strategy* pattern to be easily exchangeable and customizable by the user. Each step also has a specific interface for the implementation of external strategies for additional flexibility in case of special use cases. In the case of more general scenarios, it is possible to combine different strategies of

one step, e. g., detecting `ActionHypothesis` for known and unknown objects at the same time. For ease of use, users can request certain combinations of strategies through high-level skills, as explained in Section III-D.

D. Embedding into Memory Architecture

We fully integrate the proposed framework into the memory-centric cognitive architecture [2] implemented in ArmarX [5] where the memory acts as a mediator between the high-level capabilities of the robotic system and the low-level control components. Thus, all communication from high- to low-level passes through the robot's memory, which requires the data to have a specific format that is understandable by the memory (i. e., *IDF*). Instead of being a simple static data storage, we believe that the robot's memory should play an active role so that it can adapt to incoming multimodal and possibly associative streams of information. Our memory consists of (i) a *Working Memory* (WM) orchestrating the knowledge coming from different sources of the robot system, (ii) a *Long-term Memory* holding procedural, episodic, and semantic information persistently and (iii) a *Prior Knowledge* that is given a priori by the programmers (e. g., known grasps, object shapes, or semantic common knowledge). As part of the long-term memory, executable skills are stored in the robot's procedural memory. In the case of grasping and manipulation, a skill is represented by one or more strategies per step of our proposed framework thus creating a unique combination of strategies for one specific problem. Figure 2 and Figure 3 depict the connection between the aforementioned steps and the memory. Each step adapts its behavior based on the content of the robot's memory, e. g., object poses or common knowledge such as typical fetching and placing positions. The used parameterization as well as the final results of each step (e. g., `ActionHypothesis`, `ExecutableAction`, ...) is stored in the memory so that it can be used for debugging or offline learning. Symbolic abstractions of the skill execution result (such as success and failure) are stored in the memory as well. That way, one can trace back the complete execution status of a manipulation action.

IV. Use Cases

We show the versatility of our mobile manipulation framework through several use cases including grasping of known and unknown objects, object placing, and semi-autonomous bimanual grasping of objects.

A. Grasping of Known Objects

For the grasping of known objects the *Discovery* and *Parameterization* steps can be combined. Grasp affordances are continuously discovered based on 6D object pose estimation and manually defined grasps stored in a grasp database as part of the robot's prior knowledge. Based on those instantiated grasp hypotheses, suitable robot placements are generated in a two-step approach: First, based on our previous work [19], initial collision-free robot placements are generated. Second, a local refinement is performed by solving a non-linear optimization problem similar to [20] which considers the end-effector target pose, joint-limits avoidance, environmental collision, human-joint limits [21], [22] and maximizes the manipulability of the end-effector [23], [24] while also orienting the robot towards the object. Only if the aforementioned criteria are fulfilled to a certain extent, the action hypothesis is further considered.

The *Execution* step makes use of the referenced object pose information in the `ExecutableAction` to refine the execution pose through re-localization of the object to account for inaccuracies in previous object pose estimation, self-localization, and eventual unforeseen movement of the object itself. The action execution and movement of the end-effectors are performed as described in our previous work [18]. There, the *tool center point* (TCP) is moved from the pre-pose to the execution pose until a force threshold is reached. At this point, the `HandTrajectory`- a coordinated hand and finger motion - is executed. Afterward, the TCP is moved to a secure retract pose.

B. Grasping of Unknown Objects

In order to discover grasp affordances and to generate action hypotheses for unknown objects, we use a RGB-D camera. As shown in Figure 5, we first segment the color image using Segment Anything [25]. The `ActionHypothesis` are generated in the *Discovery* step according to [26]: object-oriented bounding boxes (OOBB) are fit to each segment in the point cloud. If the OOBB are within certain margins that conform to the robot's end-effector, grasp hypotheses are generated along the sides of the box for left- and right-handed grasps.

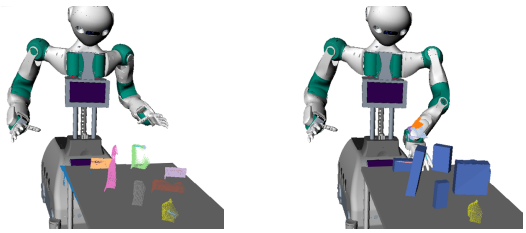

Fig. 5: Action hypothesis extraction for unknown objects. Point cloud segmentation using Segment Anything [25] (left) and object-oriented bounding boxes fitting [26] (right).

C. Object Placement at Common Places

Contextual knowledge about known objects can be added via the robot's memory to solve e. g., a pick-and-place task. This includes *common places* [27], i. e., grounded spatial symbols, that indicate where to search or where to place an object. This symbolic representation of common sense knowledge is grounded in the continuous real world in order to be useful for execution. A common place is a volumetric space defined as either absolute or relative to an object class or instance. Each object can have multiple prioritized common places which the robot can choose from in the given scene. Figure 6 shows several common locations used in

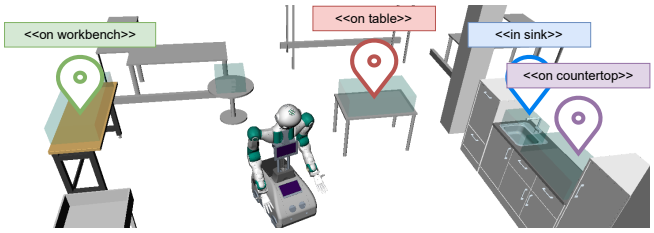

Fig. 6: Common places used in our experiments. The colored labels indicate symbolic names for each common place and the light-green boxes depict their respective position and extents in the global frame.

our experiments, including their symbolic labels and sub-symbolic real positions. Here, knowledge about common places is provided through *Prior Knowledge*, but can also be learned from experience through the episodic memory.

D. Bimanual Grasping of Unknown Objects

For the discovery of bimanual grasp affordances, an approach combining the teleoperation from [28] with the hypothesis generation of [29] can be used. The grasp candidates were generated by a human operator by clicking on a specific point in an interactive visualization of the scene during the *Discovery* step. The pose of the `ActionHypothesis` was then generated using the averaged local surface information of the point cloud by calculating the *Local Curvature Frame* at that point and defining the pose relative to that frame [29]. The pose can be adapted by the operator after the initial generation of the `ActionHypothesis`. The *Parameterization* step then only combines the two `ActionHypothesis` into one `ExecutableAction` and computes a platform placement in the centered in the middle between both hands.

During the *Execution*, a bimanual grasp candidate is treated equivalently to two independent grasp candidates by concurrently executing one grasp candidate with each arm. After each phase, the arms stop until both arms have finished the phase to synchronize the grasping process between both arms. The compliant control of both arms is done independently of each other using *Via-point Movement Primitives* [30], similar to [18]. As we do not use any form of coordination between the arms beside the four synchronization points (i. e., pre-pose, execution pose, action trajectory and retract pose), the compliant behavior of the controller helps to compensate for small misalignments of the tool center points with respect to each other when lifting or carrying an object bimanually.

V. Experiments

We performed a number of real-world experiments on the humanoid robots ARMAR-6 and ARMAR-DE to demonstrate the applicability of our framework to realistic environments. The experiments were designed to support our design decision described in Section III-A and consist of a table-clearing and box-picking scenario.

A. Clearing a Table

We show the generalization of manipulation tasks across different robots of our approach using the two humanoid robots ARMAR-6 [3] and ARMAR-DE in a table-clearing setup. Both robots are equipped with two anthropomorphic 8 degrees of freedom (DoF) arms and two underactuated five-finger hands with 2 DoF (ARMAR-6) and 4 DoF (ARMAR-DE). For 6D object pose estimation, we use SimTrack [31] on both platforms and additionally IVT [32] on ARMAR-6. Each table-clearing experiment consists of 7 different rigid and deformable objects (YCB [33] and common household objects) which are placed arbitrarily on a table in structured clutter. Each of the known objects is associated with a common place (*sink*, *kitchen countertop*, or *workbench*). As long as the robot recognizes known objects, it will prioritize manipulating them before unknown objects. Due to the aforementioned differences in 6D object pose estimation, the robots treat different objects as known and unknown. ARMAR-DE is able to recognize the *mustard*, the *bio-milk*, the *apple-tea* and the *spraybottle*. The first three objects should be placed on the *countertop* while the latter should be placed on the *workbench*. In addition, ARMAR-6 is able to recognize the *screwbox* which should be placed on the *workbench*, and the *sponge* which should be placed in the *sink*. All unknown objects or objects that the robot cannot recognize should be placed on the free table next to the kitchen as shown in Figure 6.

B. Bimanual Grasping of Unknown Objects

In addition to the table-clearing experiments, we performed a number of semi-autonomous, bimanual pick-and-place executions of larger objects on ARMAR-6 to showcase the ability of our framework to handle more than unimanual actions and incorporate user feedback through teleoperation. The bimanual grasping approach explained in Section IV-D was used for all executions.

The experimental setup was very similar to the setup from [29]: A varying number of unknown objects were placed in a box of known dimensions. The number of objects in the box was chosen to be 1, 4, 6, 8, and 10 and remained constant for 30 consecutive bimanual grasp executions. Every five grasp attempts, the scene was randomly rearranged and objects were exchanged by the operator to increase the variability of the encountered object constellations. After grasping, the object was lifted for five seconds and then placed again in the box by the robot. This additionally introduces a change in the setup and reduces the possibility of any human bias in the constructed scene.

The objects used for the box emptying experiments consist of a frying pan, an exhaust pipe, a socket strip, multiple

Fig. 7: The humanoid robot ARMAR-6 grasping and carrying multiple objects bimanually.

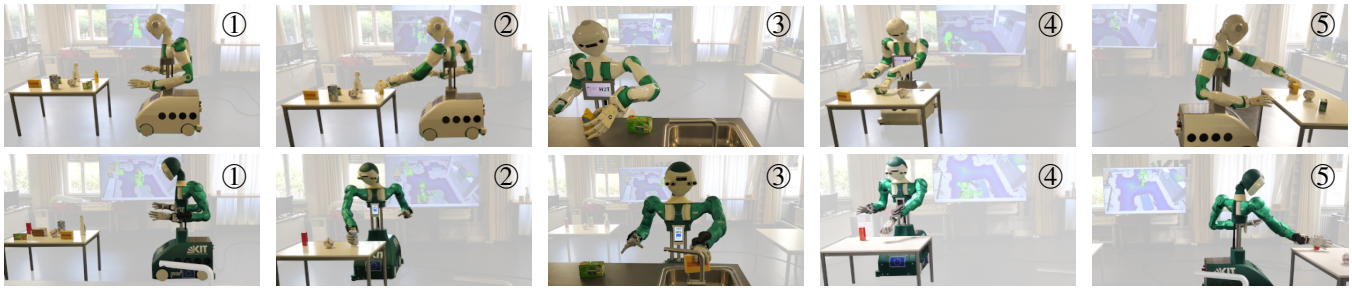

Fig. 8: Table-clearing of known and unknown objects with ARMAR-DE and ARMAR-6. ① Initial setup, ② grasping of known objects, ③ placing of known objects, ④ grasping of unknown objects, and ⑤ placing of unknown objects. An accompanying video shows the experiments.

plastic pipes, a toolbox, and a heavy spring arm of a surgical light system. During the generation of grasp candidates, we attempted to have an even distribution of grasp attempts for each object type. The results of the grasp attempts were categorized based on whether the object (1) remained off the ground for 5 seconds and is grasped firmly with both hands ("*bimanual*"), (2) was lifted for 5 seconds but is only grasped with one hand ("*unimanual*"), (3) was visibly lifted for less than 5 seconds ("*lifted*"), (4) was not lifted because of collisions with other objects or the environment ("*collision*"), or (5) was not lifted because the hand slipped off the object ("*slipped*"). Figure 7 shows exemplary successful bimanual grasp executions of different objects.

C. Discussion

Qualitative results of the table-clearing experiments are shown in Figure 8. Both robots were able to perform multiple pick-and-place attempts successfully. Failed attempts mainly originated from imprecisely generated grasp poses for known objects due to the inaccuracy in the 6D object pose estimation and wrongly generated object-oriented bounding boxes for unknown objects.

The results of the box emptying experiments can be seen in Figure 9 according to the categorization described in Section V-B. The average success rate where at least one grasp was successful and the object was lifted (categories "*bimanual*" and "*unimanual*") is 77.3%. However, this rate drops to 44.7% when only bimanual grasps are counted. When dealing with a single object in the box, our teleoperated approach demonstrated high precision, with an 83.3% success rate for bimanual grasps. This large drop in performance is to a large degree caused by collision with other objects. Performing a bimanual grasp implies that the chance for collision of a hand with other objects is twice as high as in the unimanual case. This is clearly visible in the results. As soon as multiple objects were in the box, the success rate of bimanual grasps dropped significantly.

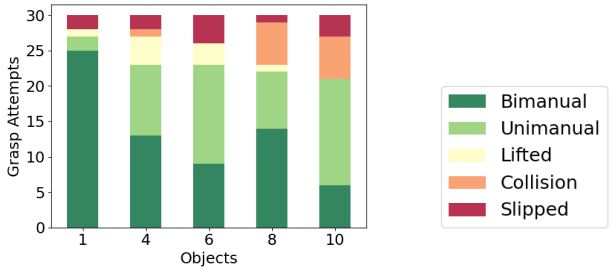

Fig. 9: Results of the box emptying experiments from Section V-B. A total of 30 grasp attempts were performed for each degree of clutter in the box.

VI. Conclusion

With this work, we presented a modular, memory-centered, and affordance-based grasping and manipulation framework for multi-handed, mobile robots, unifying the autonomous manipulation process for known and unknown objects in arbitrary environments. In two complex real-world experiments, we showed that our framework (I) can be used for different task definitions, such as grasping known and/or unknown objects, (II) can be applied to different robots with different kinematics (i. e., hands), (III) supports multiple autonomy levels (full autonomy and teleoperation), and (IV) can be used for the execution of uni- and bimanual actions. Additionally, we showed that the link to a cognitive memory offers contextual awareness, supporting the utilization of common knowledge to enhance the manipulation process but also facilitating learning from both success and failure. Technical explanations, such as the decomposition into five distinct steps (*Discovery*, *Parameterization*, *Validation*, *Selection*, *Execution*), related data types, and the integration into the cognitive architecture of our robots, provide a deep insight into the overall system.

At this point in time, the presented framework is focused on grasp and placement affordances. We plan to extend our approach to additional affordance types, e. g., for opening and closing of drawers and doors. To account for failures, especially during grasping attempts, we will combine our framework with more reactive mobile manipulation approaches. In order to perform a reproducible quantitative evaluation of the overall framework, we plan to standardize an evaluation scenario including grasping of known and unknown objects in highly cluttered scenes.

Acknowledgements

We would like to thank Rainer Kartmann, Patrick Hegemann, Abdelrahman Younes, and Andre Meixner for their contributions, support, and assistance during the development of this framework.

References

[1] J. J. Gibson, "The theory of affordances," in *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979, ch. 8, pp. 119–137.

[2] F. Peller-Konrad, R. Kartmann, C. R. G. Dreher, A. Meixner, F. Reister, M. Grotz, and T. Asfour, "A Memory System of a Robot Cognitive Architecture and its Implementation in ArmarX," *Robotics & Autonomous Systems*, vol. 164, pp. 1–20, 2023.

[3] T. Asfour, M. Wächter, L. Kaul, S. Rader, P. Weiner, S. Ottenhaus, *et al.*, "ARMAR-6: A High-Performance Humanoid for Human-Robot Collaboration in Real World Scenarios," *IEEE Robotics & Automation Magazine*, vol. 26, no. 4, pp. 108–121, 2019.

[4] M. Wächter, S. Ottenhaus, M. Kröhnert, N. Vahrenkamp, and T. Asfour, "The ArmarX Statechart Concept: Graphical Programming of Robot Behaviour," *Frontiers in Robotics & AI*, vol. 3, pp. 0–0, 2016.

[5] N. Vahrenkamp, M. Wächter, M. Kröhnert, K. Welke, and T. Asfour, "The robot software framework ArmarX," *it - Information Technology*, vol. 57, 2015.

[6] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *IEEE International Conference on Robotics and Automation, Workshop on Open Source Robotics*, 2009.

[7] F. Rovida and V. Kruger, "Design and development of a software architecture for autonomous mobile manipulators in industrial environments," in *IEEE International Conference on Industrial Technology (ICIT)*, 3 2015, pp. 3288–3295.

[8] Y. Jiang, N. Walker, M. Kim, N. Brissonneau, D. S. Brown, J. W. Hart, S. Niekum, L. Sentis, and P. Stone, "LAAIR: A layered architecture for autonomous interactive robots," *CoRR*, vol. abs/1811.03563, 2018.

[9] S. Hart, P. Dinh, and K. Hambuchen, "The Affordance Template ROS Package for Robot Task Programming," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 6227–6234.

[10] S. Hart, A. H. Quispe, M. W. Lanighan, and S. Gee, "Generalized Affordance Templates for Mobile Manipulation," in *IEEE International Conference on Robotics and Automation*, 2022, pp. 6240–6246.

[11] A. Hermann, Z. Xue, S. W. Rühl, and R. Dillmann, "Hardware and Software Architecture of a Bimanual Mobile Manipulator for Industrial Application," in *IEEE International Conference on Robotics and Biomimetics*, 2011, pp. 2282–2288.

[12] P. Nebot and E. Cervera, "An integrated agent-based software architecture for mobile and manipulator systems," *Robotica*, vol. 25, pp. 213–220, 3 2007.

[13] J.-B. Yi, T. Kang, D. Song, and S.-J. Yi, "Unified Software Platform for Intelligent Home Service Robots," *Applied Sciences*, vol. 10, no. 17, p. 5874, 2020.

[14] T. Keleştemur, N. Yokoyama, J. Truong, A. A. Allaban, and T. Padir, "System Architecture for Autonomous Mobile Manipulation of Everyday Objects in Domestic Environments," in *International Conference on Pervasive Technologies Related to Assistive Environments*. ACM, 2019, pp. 264–269.

[15] J. A. Bagnell, F. Cavalcanti, L. Cui, T. Galluzzo, M. Hebert, M. Kazemi, M. Klingensmith, J. Libby, T. Y. Liu, N. Pollard, M. Pivtoraiko, J.-S. Valois, and R. Zhu, "An Integrated System for Autonomous Robotics Manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, p. 2955–2962.

[16] M. Logothetis, G. C. Karras, S. Heshmati-Alamdari, P. Vlantis, and K. J. Kyriakopoulos, "A Model Predictive Control Approach for Vision-Based Object Grasping via Mobile Manipulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 10 2018, pp. 1–6.

[17] B. Burgess-Limerick, C. Lehnert, J. Leitner, and P. Corke, "An architecture for reactive mobile manipulation on-the-move," in *IEEE International Conference on Robotics and Automation*. IEEE, 2023, pp. 1623–1629.

[18] C. Pohl, P. Hegemann, B. An, M. Grotz, and T. Asfour, "Humanoid Robotic System for Grasping and Manipulation in Decontamination Tasks," *at - Automatisierungstechnik*, vol. 70, pp. 850–858, 2022.

[19] F. Reister, M. Grotz, and T. Asfour, "Combining navigation and manipulation costs for time-efficient robot placement in mobile manipulation tasks," *IEEE Robotics & Automation Letters*, vol. 7, no. 4, pp. 9913–9920, 2022.

[20] D. Rakita, H. Shi, B. Mutlu, and M. Gleicher, "Collisionik: A per-instant pose optimization method for generating robot motions with environment collision avoidance," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 9995–10 001.

[21] K. Luttgens, H. Deutsch, and N. Hamilton, *Kinesiology: scientific basis of human motion*. Brown & Benchmark, 1992.

[22] C. Gäbert, S. Kaden, and U. Thomas, "Generation of human-like arm motions using sampling-based motion planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 2534–2541.

[23] T. Yoshikawa, "Manipulability of robotic mechanisms," *The international journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[24] J. Haviland, N. Sünderhauf, and P. Corke, "A holistic approach to reactive mobile manipulation," *IEEE Robotics & Automation Letters*, vol. 7, no. 2, pp. 3122–3129, 2022.

[25] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[26] R. Grimm, M. Grotz, S. Ottenhaus, and T. Asfour, "Vision-based robotic pushing and grasping for stone sample collection under computing resource constraints," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 6498–6504.

[27] K. Welke, P. Kaiser, A. Kozlov, N. Adermann, T. Asfour, M. Lewis, and M. Steedman, "Grounded spatial symbols for task planning based on experience," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2013, pp. 484–491.

[28] P. Kaiser, D. Kanoulas, M. Grotz, L. Muratore, A. Rocchi, E. M. Hoffman, N. G. Tsagarakis, and T. Asfour, "An Affordance-based Pilot Interface for High-Level Control of Humanoid Robots in Supervised Autonomy," in *IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 621–628.

[29] C. Pohl and T. Asfour, "Probabilistic Spatio-Temporal Fusion of Affordances for Grasping and Manipulation," *IEEE Robotics & Automation Letters*, vol. 7, no. 2, pp. 3226–3233, 2022.

[30] Y. Zhou, J. Gao, and T. Asfour, "Learning Via-Point Movement Primitives with Inter- and Extrapolation Capabilities," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 4301–4308.

[31] K. Pauwels, L. Rubio, and E. Ros, "Real-time Pose Detection and Tracking of Hundreds of Objects," *IEEE Transactions on Circuits and Systems for Video Technology*, 2015.

[32] P. Azad, T. Asfour, and R. Dillmann, "Stereo-based 6D Object Localization for Grasping with Humanoid Robot Systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 919–924.

[33] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *International Conference on Advanced Robotics (ICAR)*, 2015, pp. 510–517.