A Modular Framework for Knowledge-Based Servoing: Plugging Symbolic Theories into Robotic Controllers

Malte Huerkamp[1,*,a], Kaviya Dhanabalachandran[1,*,b], Mihai Pomarlan[2,c], Simon Stelter[1,d] and Michael Beetz[1,e]

[1]*Institute for Artificial Intelligence, University of Bremen, Germany*

[2]*Applied Linguistics Department, University of Bremen, Germany*

{*huerkamp, kaviya, pomarlan, stelter, mbeetz*}*@uni-bremen.de*

Keywords: Symbolic Reasoning, Motion Control, Robotic Manipulation.

Abstract: This paper introduces a novel control framework that bridges symbolic reasoning and task-space motion control, enabling the transparent execution of household manipulation tasks through a tightly integrated reasoning and control loop. At its core, this framework allows any symbolic theory to be ”plugged in” with a reasoning module to create interpretable robotic controllers. This modularity makes the framework flexible and applicable to a wide range of tasks, providing traceable feedback and human-level interpretability. We demonstrate the framework using a qualitative theory for pouring with a defeasible reasoner, showcasing how the system can be adapted to variations in task requirements, such as transferring liquids, draining mixtures, or scraping sticky materials.

1 INTRODUCTION

In robotics, the demand for systems that are both transparent and interpretable has become increasingly essential, particularly in safety-critical applications. Although data-driven methods, including multimodal foundation models for robot control, have demonstrated exceptional performance in tasks that require generalization and semantic reasoning, their decision-making processes remain opaque (Brohan et al., 2023). This lack of transparency complicates introspection, debugging, and maintenance. Such challenges are especially concerning in dynamic settings where robots must operate safely, reliably and adaptively, with decision-making processes that are traceable and comprehensible, such as in household environments.

At the same time, we challenge the common view that qualitative inference is merely a compromise for achieving higher-order goals like interpretability. While quantitative precision remains important, human skill acquisition suggests that qualitative, symbolically describable knowledge plays a fundamental role in mastering complex motor tasks. Symbolic descriptions not only support learning through language and communication but also serve as a means to abstract and transfer knowledge across tasks and environments. By embracing qualitative inference, we enable robots to adapt to task variations and environmental changes more robustly, a property often lacking in purely data-driven systems. For example, while reinforcement learning agents can achieve superhuman mastery in specific games, their policies often fail when faced with even trivial changes in the game environment (Kansky et al., 2017) — highlighting the importance of qualitative, causal knowledge for generalization.

This paper introduces a novel control framework that bridges symbolic reasoning and full-body robot motion control, enabling the interpretable execution of household manipulation tasks through tightly integrating symbolic reasoning into a motion control loop. At its core, the framework enables a variety of symbolic theories, that satisfy certain requirements, to be ”plugged in” with a reasoning module into a motion controller to create interpretable robotic controllers. The requirements will be outlined in this paper. Once connected, the reasoning module generates actionable decisions, while the control system exe-

[a] https://orcid.org/0009-0008-5880-6484
[b] https://orcid.org/0000-0002-0419-5242
[c] https://orcid.org/0000-0002-1304-581X
[d] https://orcid.org/0000-0002-0066-1904
[e] https://orcid.org/0000-0002-7888-7444
*These authors contributed equally to this work.

Huerkamp, M., Dhanabalachandran, K., Pomarlan, M., Stelter, S. and Beetz, M.
A Modular Framework for Knowledge-Based Servoing: Plugging Symbolic Theories into Robotic Controllers.
DOI: 10.5220/0013394400003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 1*, pages 886-897
ISBN: 978-989-758-737-5; ISSN: 2184-433X

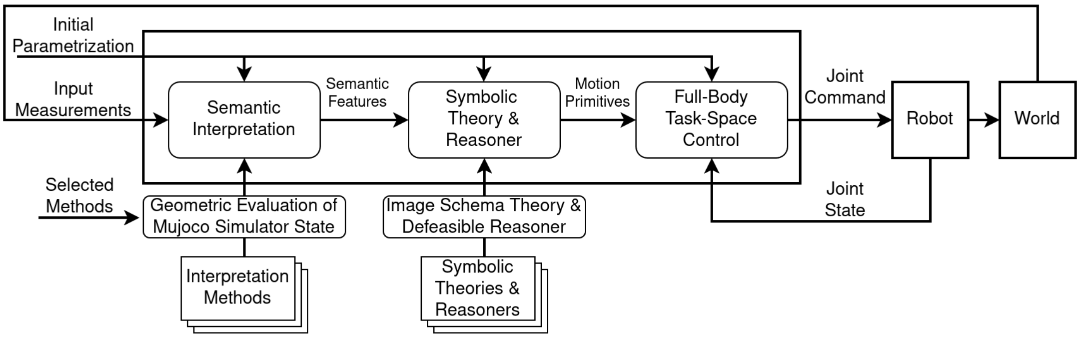

Figure 1: Conceptual overview of the knowledge-based servoing framework and the semantic interpretation and reasoning methods we selected for the examples in the evaluation section.

cutes them in real time. This modularity makes the framework applicable to a wide range of tasks, providing traceable feedback, human-level interpretability, and a opportunity for robust knowledge transfer between tasks and environments.

We demonstrate the framework using a symbolic theory for pouring, based on a concept from cognitive science called image schemata (Mandler, 1992). Image schemata, as dynamic patterns of spatial relations and movements, provide a vocabulary to describe situations at an abstract level while capturing functionally relevant aspects, such as containment, support, or linking of objects. By leveraging image schemata, our understanding of spatial arrangements in terms of the affordances they enable, prevent, or manifest, and how relative movements between objects impact task goals can be expressed as a set of rules on symbolic predicates. The rule-based formalism we use for that is called defeasible logic (Antoniou et al., 2000). Furthermore, we apply the basic rule set developed for pouring, on the tasks of draining mixtures and scraping sticky materials by adapting the initial rule set based on our understanding of the new task. However, the framework is not limited to pouring or defeasible logic. By design, it supports the integration of multiple symbolic theories, making it extensible to diverse tasks and environments. Through this plug-and-play capability, our framework promotes a systematic approach to developing transparent robotic controllers for complex, real-world applications.

The contributions of this paper are:

1. A modular control framework that integrates symbolic reasoning and control, enabling the execution of tasks through pluggable symbolic theories.
2. A demonstration of the framework's flexibility and transparency using a rule-based theory derived from image schemata concepts to reason about pouring processes.
3. An evaluation of the system's adaptability across robots, environments, and task variations, highlighting trade-offs between performance and transparency.

This work establishes a foundation for knowledge-based servoing, a paradigm inspired by visual servoing but extended to symbolic reasoning. By combining qualitative inference with real-time execution, the proposed framework empowers robots to perform complex tasks with transparency, adaptability, and robust generalization, while facilitating the transfer of knowledge across tasks and environments.

The next section explains the basic idea and the assumptions we made for different parts of the knowledge-based servoing framework. Section 3 presents related work and section 4 explains the reasoning method used for the pouring example. Afterwards, section 5 details the integration of symbolic reasoning with a task space control method. Then we present the evaluation of different pouring tasks in section 6 and end with the conclusion in section 7.

2 THE KNOWLEDGE-BASED SERVOING PARADIGM

Visual servoing in robotics uses visual feedback to guide movements by connecting observed image features to control parameters through a mathematical model (Chaumette and Hutchinson, 2006). While effective for precision tasks, it is limited by its reliance on visible image features and struggles due to occlusions in partially observable environments.

In knowledge-based servoing, on the other hand, we want to operate on semantic features (also called facts) of the environment, the task, and the robot. Semantic features can be extracted from, but are not limited to, vision. Other sensor modalities such as tactile or force/torque sensors can also be used. Different modalities can provide a measurement for the same semantic feature in case one modality is unavailable, e.g. due to occlusion. In the example of pouring, a simple semantic feature would be that the destination container is not filled to the desired level, therefore pouring has to continue.

To achieve knowledge-based servoing, real-time reasoning preceded by a suitable semantic interpretation layer has to be integrated into the control loop. Figure 1 shows a diagram of the proposed architecture. The semantic interpretation layer filters all input modalities for the facts that are relevant for the used symbolic theory. The theory is then evaluated based on the perceived facts and the results are forwarded to a full-body task-space controller that moves the robot accordingly. As in visual servoing, where there exist multiple methods for feature detection, the design of the interaction matrix, or the direct calculation of image differences, it should be possible to use a variety of symbolic theories in knowledge-based servoing. More specifically, we want to design the integration in the control loop in a way that supports a "plug-and-play" like change of symbolic theories from a library that is part of the robot's knowledge base. To realise that, assumptions and requirements for each module of the control cycle have to be defined and satisfied.

2.1 Assumptions and Requirements

First, for any symbolic theory to be applicable, a set of input values i.e. facts has to be defined once for the specific symbolic theory and then grounded in every step of the control cycle. Therefore, as usual in visual servoing, the control cycle starts with perceiving data that are then semantically interpreted into relevant facts. The discretized information is then analyzed by the symbolic reasoner based on the provided theory to infer the desired movement of the robot. Because of the discretization of the data and the absence of a concrete mathematical model, it is not possible to calculate the desired movement in terms of a continuously valued output value. Instead, the reasoner has to infer a set of motion primitives that the motion controller has to realize. This has the advantage that motion primitives could be arbitrary complex, if the employed motion controller supports them. For the sake of generality, we propose to choose motion primitves that can be transformed into a desired task frame twist. Second, the symbolic theory has to be solvable sufficiently fast by the used reasoner. The same holds for the semantic interpretation of perceived data.

The dependence on motion primitives requires the controller to be able to execute each primitive alone or as a composition of multiple ones. It also has to be able to smoothly switch from one primitive to another in one control cycle. A class of motion controllers that can to do this are task-space control frameworks that solve a quadratic optimization problem for instantaneous joint velocities or torques (Mansard et al., 2009), (Aertbeliën and De Schutter, 2014), (Bouyarmane et al., 2018), (Stelter et al., 2022a), (Corke and Haviland, 2021), (Escande et al., 2014). This way, the combination of all active motion primitives can be represented as a desired task frame twist that can change abruptly at every control cycle. A desired twist can be defined for multiple task frames to realize dual arm manipulation or to control the field of view of head mounted cameras. They can also be combined with other tasks to enforce safety constraints like joint limits and collision avoidance.

3 RELATED WORK

Symbolic inference methods have been used in robotics mostly for aspects of behavior that correspond to high levels of abstraction, such as task planning; a survey on the state of the art, with a focus on declarative and logic based methods, is given in (Meli et al., 2023). Because robot behaviors must take into account both constraints at higher levels of abstraction as well as constraints imposed by geometry and physics, the field of Task And Motion Planning (TAMP) is very active, and a recent survey is provided by (Guo et al., 2023). Outside of planning, logic-based methods have been employed either to describe "controller" specifications – "controller" here meaning a state machine guiding paths in a discrete transition system – as in (Kress-Gazit et al., 2011) or to specify rules with which to select a next action as in (Xiao et al., 2021; Lam and Governatori, 2013; Ferretti et al., 2007; Shanahan and Witkowski, 2001); several of these papers even present applications of defeasible logic in robotics. In general, in the cited papers, inference operates on highly abstracted, atomic actions loosely coupled with what lower-level control actually does. A more direct connection between symbolic logic and control is explored in (Lindemann and Dimarogonas, 2019) where specifically designed temporal properties of control barrier functions are used to satisfy signal temporal logic tasks. Their design requirements limit the set of feasible signal temporal logic expressions, while we aim for an approach where more general symbolic theories can be used. For that reason the evaluations of formal guarantees for our control system, as it is done in the field of formal synthesis, is outside the scope of this paper.

A framework with a stronger focus on robotics applications is presented in (Muhayyuddin et al., 2017), where ontologies and physics based motion planning are combined with linear temporal logic (LTL) specifications. The knowledge-based framework can consider the capabilities of the robot during LTL verification and physics-based motion planning to realize the planning of long-horizon tasks that might require the manipulation of obstacles. In contrast, we are focused on the control of local manipulation tasks rather than long horizon tasks and motion planning. Still, we see great benefit in embedding our framework in an overarching planning system for the initial parameterization based on given task requests, especially if digital twins of the environment and the robot are used to evaluate the outcome of different parameterizations beforehand. Here, the symbolic theory embedded in the controller provides a benefit in automatic debugging of the robots behavior. But this is outside the scope of this paper.

In regards to the closed-loop control of our running example, the pouring task, significant advances have been made in pouring liquids using tactile information (Piacenza et al., 2022) or vision (Zhu et al., 2023), (Schenck and Fox, 2017), (Dong et al., 2019). However, these methods primarily focus on adjusting the tilt angle based on fill-level feedback. This is in contrast to our work, which is a control system capable of accommodating various forms of feedback. In the case of pouring, this includes the fill level of participating containers, if spilling has occurred, and if the placement of the containers allows pouring. To the best of our knowledge, there exists no motion control system that includes these types of feedback in closed-loop pouring control while also considering the tasks for draining of mixtures and scraping of sticky materials. Motion planning for pouring with a fluid dynamics model in (Pan et al., 2016) implicitly considers the placement between containers, but lacks real-time capabilities due to the high computational demand of the fluid model.

4 SYMBOLIC THEORIES PLUGGABLE INTO ROBOT CONTROL

In this section we give an example of a suitable symbolic theory for our framework. Therefore, we outline the contents of a theory that qualitatively reasons about the physics of pouring, and the inference method we have chosen for the evaluation examples in this paper.

4.1 Qualitative Theories of Physics

To be applicable to the knowledge-based servoing framework, a theory has to operate on qualitative facts asserted by perception, infer high level descriptions of the situation, and activate or suspend motion primitives that are then converted to desired twist values. While perception and control modules are then tasked to interface the quantitative world with the qualitative, we now turn to the contents of this qualitative representation.

The relevant level of abstraction at which a theory for knowledge-based servoing should operate is that of the presence/absence and (im)possibility of motion between objects. For the constitution of such a theory, we have used notions from cognitive science: image schemas and affordances.

Image schemas are suggested in cognitive linguistics to play a complex role: on the one hand, they are sensorimotor patterns of embodied experience, on the other they abstract away from quantitative details while preserving functional aspects of a scenario (Johnson, 1987). Examples of image schemas include Linkage, Containment, Support. Thus, image schemas provide us with a vocabulary with which to describe object interactions, at a level where one can answer questions such as, are objects moving apart from each other, can they move apart from each other, will they move together if one of them moves etc.

To illustrate how image schemas may describe a timeline of events, consider a prototypical pouring scenario shown in Figure 2, which proceeds through a sequence of scenes characterized in qualitative terms. Particulars of shape or coordinates are abstracted away, but the various steps along the sequence – separated by differences in which image schematic relations are present – capture the functionally relevant aspects of pouring. The image schematic description contains information about necessary conditions for something to happen. In the example, for the contents to exit or enter a container (scene 4) it must pass through the container's boundary, and this boundary must not be blocked for this to be possible. The image schematic description implies expectations of observed behavior. In the figure, because the contents are contained (scenes 1, 2), we expect it to move together with the container (scene 2). Such expectations may fail to materialize for reasons yet invisible to the robot, but they are important to indicate what a default course of events would be, and thus guide perception and monitoring systems to select what kinds of queries are relevant to answer.

While the above focuses more on object interaction, affordances bring the robot and its actions into

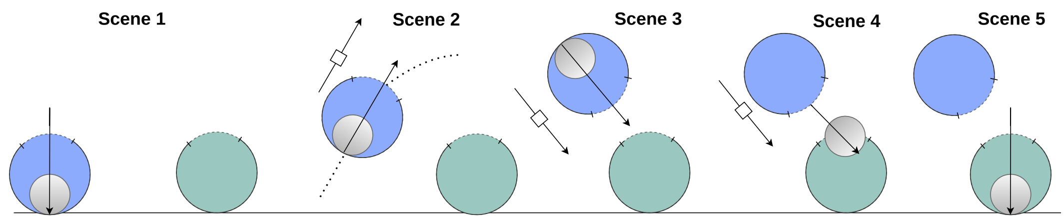

Figure 2: Image schematic segmentation of a pouring event. Scene1: Substance is contained inside source and the source affords CONTAINMENT. Both source and destination are FAR and VERTICAL. Scene2: Source MOVE UP and NEAR destination. Source is not VERTICAL anymore. The arrow with □ indicates a caused movement. Scene3: Source MOVE TOWARDS destination and there is no BLOCKAGE. Scene4: Substance goes OUT of source and IN to the destination. Substance has a SOURCE-PATH-GOAL. Scene5: Substance is contained inside destination and the destination affords CONTAINMENT.

the picture. Affordances are what an environment provides to an agent in terms of possibilities for action (Gibson, 1977). Reasoning with affordances enables answering questions related to what actions are possible, what their effects would be, and what other objects i.e. tools should be involved in the action.

Our theories then infer image schematic descriptions based on qualitative facts asserted by perception, and in so doing infer expectations about how these objects can and will act. They infer affordances based on the image schematic description, select or tune motion primitives based on the inferred affordances, and emit queries to perception to verify that expectations implied by the image schematic descriptions are met.

Using square brackets to index by discrete time steps, the integration of reasoning into the larger perception-control loop and its embedding into the world can then be summarized as:

$$\begin{aligned} X[k+1] &= ENV(X[k], U[k]) \\ Y[k+1] &= SEMINT(X[k+1], Q[k]) \\ S[k] &= SCHMOD(Y[k]) \\ U[k] &= INVMOD(S[k], G) \\ Q[k] &= FWDMOD(S[k], U[k], G) \end{aligned}$$

In the above, X is the quantitative state of the world, U a description of which motion primitives are active for the robot, and ENV is a function that quantitatively updates the state of the environment. Y are qualitative facts about observed object movements and spatial relations, Q is a description of perception queries to run, and SEMINT is a function from environment state to qualitative observations. Each of SCHMOD, INVMOD, FWDMOD are collections of rules. SCHMOD infers image schemas and affordances from observed qualitative facts. INVMOD acts as a physics inverse model to infer the motion primitives to execute. FWDMOD acts as a physics forward model to infer what to expect and thus what to query for in the next time step. In the above, G stands for a set of facts characterizing the robot's overarching goal for the task, here assumed to be stable for the duration of the task.

Facts that need to be inferred for the theory we employ in this paper are the relative poses of two containers and geometric reasoning about how close they and their openings are; if there is outflow from the source container; if there is spilling; is the pouring goal is reached. The set of motion primitives is {moveLeft, moveRight, moveUp, moveDown, moveForward, moveBack, increaseTilting, decreaseTilting, rotateLeft, rotateRight}.

4.2 Defeasible Inference

As qualitative reasoning leaves detail out, its conclusions will not be always true. Cups can contain water, except when they cannot – because of being cracked or turned upside down, etc. Therefore, a robot must always watch what the world actually does and react accordingly. However, it is also helpful to represent and reason with exceptional cases when these are known. A logical formalism which allows this and does so efficiently is defeasible logic (Antoniou et al., 2000). It allows inferring what would typically be thought true in a situation but allows retraction of conclusions when additional information is given. A defeasible theory is represented by (R, >), with R being a finite set of defeasible rules and '>' a superiority relation among the rules. Defeasible inference proceeds by adding a conclusion to a provability chain when there is no undefeated objection to that conclusion remaining in the theory. Contradictory conclusions (e.g. p and its negation, denoted $-p$) object to each other. An objection is defeated when all the rules supporting it are inapplicable or overruled by superior applicable rules. A rule is applicable if all of the terms in its condition are facts or have already been added to the provability chain.

A simple example of a defeasible theory which showcases the default and exceptions pattern is given below:

$$r : Container(?x) \Rightarrow canContain(?x)$$
$$s : Cup(?x) \rightarrow Container(?x)$$
$$r' : Broken(?x) \Rightarrow -canContain(?x)$$
$$r' > r$$

"$\Rightarrow$" represents a defeasible implication, i.e. a conclusion that could be retracted upon further information, but seems the best one given the available data right now. In the rules, variable names begin with a '?'. Binding variables in a rule to entities in a robot's situation grounds the rule, and defeasible inference will proceed only on grounded rules. The theory states that Containers can contain, Cups are Containers but a broken cup cannot contain.

We then constructed a defeasible rule set to encode a qualitative theory for pouring in a general way. As an example of a rule about what can happen and what can we expect to observe, consider a scenario in which the affordance to pour is met and the tilt motion has been carried out. According to the image schematic sequence of events during pouring, the contents will be out. This expectation about the state is defined as an attention query to check if the contents are out.

$$\text{Source}(?s), \text{Destination}(?d), \text{canPourTo}(?s, ?d),$$
$$\text{isTilted}(?s) \Rightarrow \text{Query\_contentsOut}(?s, ?d)$$

If the contents are not out from the source as expected then the concluded motion primitive for the controller is to react by increasing the tilting of the source.

$$\text{Source}(?s), \text{Destination}(?d), \text{canPourTo}(?s, ?d),$$
$$\text{isTilted}(?s), \text{-contentsOut}(?s, ?d)$$
$$\Rightarrow \text{Perform\_IncTilting}(?s)$$

By reasoning about expected facts the perception module could be optimized to not calculate facts that are not expected, which would result in intelligently deciding when to utilize which sensor for what purpose.

5 DESIGN OF THE MOTION CONTROL METHOD

To link the output of the reasoner to a task space control method, the higher level planning component that selects the symbolic theory and initializes the reasoner, also has to initialize a task space control interface with the correct structure to interpret the feedback provided by the reasoner. For the example of pouring this includes defining the tilt direction $\mathbf{n}_t \in \mathbb{R}^3$, the rotation direction for rotating a container around its height axis $\mathbf{n}_z \in \mathbb{R}^3$, velocity gain parameters (α, β, γ), which task frame should be controlled, and a common reference frame.

From a mathematical perspective, the reasoner provides a set of motion primitives where each primitive corresponds to a movement along or about an axis of a reference frame summarized as a set of Boolean values $B = \{x_+, y_+, z_+, x_-, y_-, z_-, t_+, t_-, r_+, r_-\} \in [0, 1]$. For example, x_+ indicates that the controlled task frame should move along the x-axis of the common reference frame in the positive direction, t_+ indicates that the task frame should rotate about the positive tilt direction transformed into the common reference frame, and r_+ indicates the same for the positive rotation about the rotation direction. The desired tool frame twist $\xi \in \mathbb{R}^6$ received from the reasoning component is constructed as:

$$\xi_d = \begin{pmatrix} \mathbf{v}_d \\ \omega_d \end{pmatrix} = \begin{pmatrix} \alpha(x_+ - x_-) \\ \alpha(y_+ - y_-) \\ \alpha(z_+ - z_-) \\ \mathbf{n}_t(\beta t_+ - \gamma t_-) + \mathbf{n}_z \beta(r_+ - r_-) \end{pmatrix} \tag{1}$$

where $\mathbf{v}_d, \omega_d \in \mathbb{R}^3$ are the desired translational and rotational velocity, respectively.

The desired tool frame twist is then integrated as a constraint in a quadratic problem of the general form

$$\begin{aligned} \min_{\mathbf{s}} \quad & \mathbf{s}^T \mathbf{H} \mathbf{s} \\ \text{s.t.} \quad & \mathbf{l}_A < \mathbf{A}\mathbf{s} < \mathbf{u}_A \\ & \mathbf{l} < \mathbf{s} < \mathbf{u} \end{aligned} \tag{2}$$

Details are presented in (Stelter et al., 2022b) where the motion control method we employ here is explained, but in a nutshell, $\mathbf{s} = (\dot{\mathbf{q}}, \mathbf{c})$ are the robot's instantaneous joint velocities and slack variables for constraint relaxation, respectively. $\mathbf{H}$ is a diagonal weight matrix describing the importance of the joints relative to each other and to the slack variables. The slack variable weights describe how expensive it is to violate their corresponding constraints. $\mathbf{A}$ contains the Jacobians of the task spaces. In this scenario, the task space describes the task frame pose with respect to the common reference frame. This makes $\mathbf{As}$ the task space velocity, i.e., the task frame twist with our chosen task space. $\mathbf{A}$ also adds one slack variable to each constraint to allow the solver to violate constraints, this is important to avoid infeasibility. $\mathbf{l_A}$ and $\mathbf{u_A}$ contain the lower and upper limits for the task space velocity, i.e., ω_d in our example. $\mathbf{l}$ and $\mathbf{u}$ contain joint velocity limits.

6 EVALUATION

In this section, we evaluate the utility of our framework in the context of pouring and variations of that task. Pouring serves as an ideal example of the framework's pluggability, as each task variation—rooted in a single abstract concept—introduces unique requirements while preserving transferable task knowledge. This highlights the efficiency of only adapting the decision-making process by plugging in different symbolic theories rather than developing similar control structures for each task. The task variations we investigate are pouring from one container to another, draining one substance from a pot while retaining another, and scraping sticky objects from a cutting board. In the standard pouring task, we evaluate whether our proposed system works at all, discuss accuracy and performance trade-offs, and show the interpretability of our control system by evaluating the symbolic theory at different snapshots of the task execution.

In the second experiment, we show how altering the symbolic theory based on the human understanding on how a task should be solved enables our control system to solve a novel task variation.

In the third experiment, we extend the theory to more motion primitives and the control structure to two controlled task frames, to showcase the extended applicability of our framework. The experiments are assessed in a simulated environment to ensure that vision algorithms for sensing container fill levels and spillages do not become the limiting factor in our evaluation. The related work section has discussed works that do this and future work will investigate how we can integrate their solutions for perception in the real world. In the following, we briefly introduce the general setup and then discuss the individual experiments in detail.

6.1 Experimental Setup

As a simulation environment, we use Mujoco with models of the bimanual mobile robot PR2 and the one-armed mobile Human Support Robot (HSR) from Toyota. The liquid in the simulated scenes is approximated by adding particles with a radius of 0.5cm to the source containers. The task-space controller and the reasoner are configured to run with a control frequency of 50hz and 10hz, respectively. The reasoner could run with a significantly higher frequency but it has to run slower than the task-space controller for a stable control loop. The velocity gains in (1) are set to $\alpha = 0.02, \beta = 0.03, \gamma = 1$ for all subsequent experiments.

6.2 Pouring Between Containers

For pouring between containers, we created a Mujoco scene in which two cups are placed on a table, Figure 3. The PR2 and HSR then grasp the cup filled with particles with one hand. Then the proposed control system is initialized and the desired amount of particles is poured into the other cup. During initialization, the system is parameterized to act on a coordinate frame at the center of the grasped cup, and the reasoner receives information about the action (pouring), the relevant objects (two cups), and the goal (to fill the destination with 40 particles).

Snapshots in Figure 3 shows the critical stages of the pouring task and how the reasoner and the controller handle them together. In the initial stage on the left, the source cup is held upright above the destination cup. The openings of the source cup and the destination cup are not yet arranged properly, and the destination is to the left of the source. Therefore, the reasoner concludes to command the motion primitive moveRight to progress toward satisfying the initial condition for pouring, that is, to align the opening of the source container with respect to the destination container.

In the second picture, the first cup is already pouring into the second cup. The reasoner is aware of this and is therefore observing the particles and the fill level of the destination container with corresponding queries to the semantic interpreter. As the cup is tilted and the particles are moving out, the reasoner observes a slow flow of the contents and hence concludes that the source cup has to be tilted more to accelerate the pouring action.

In the last picture, the first cup is tilted back to stop pouring. This happens because the affordance to pour is no longer needed as the desired goal state is achieved. In addition to that, the reasoner concludes decreaseTilt until the cup is upright. When the cup is upright and the goal is reached, the reasoner will not activate any motion primitive to indicate the end of the task execution.

We executed this experiment 60 times with different goal conditions, starting positions of the source cup around the destination cup, and different sliding friction coefficients for the particles. See Table 1 for the results. The tilt direction was automatically determined as either a leftward or rightward tilt from the gripper's perspective, depending on whether the source cup was located to the right or left of the destination cup. All runs were successful, but on average the system always overshot the desired amount of particles. The most overshoot happened for low goals, as the particles tend to come out in a bulk and our system

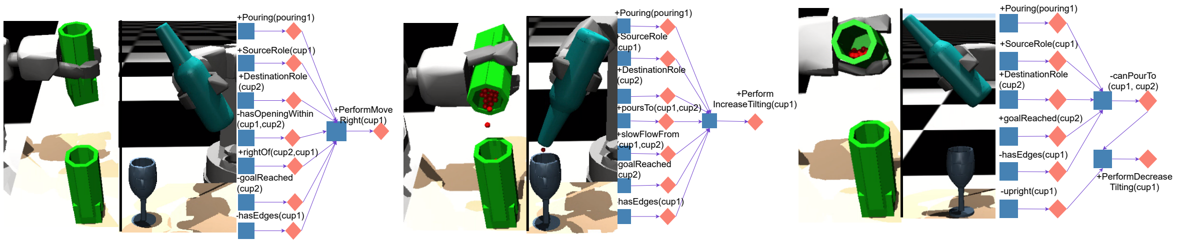

Figure 3: Three different stages of a pouring task with the reasoner's inference for that stage. In the reasoning graph, boxes indicate rules and diamonds refer to the inferred predicates. The shown graphs are curated snapshots from the full evaluated state of the defeasible reasoner to show the inference process of a specific movement primitive. Normally, multiple movement primitives can be inferred at the same time. The reasoning process is the same for pouring from a bottle to a wineglass or from a cup to a cup.

will react a touch too late to stem the flow, once this flow is observed. In contrast, state of the art liquid volume estimation in combination with a PID controller for the rotation of a source container around one axis achieves final goal errors of below one percent (Zhu et al., 2023). This is a significant difference from our best performing scenarios of pouring 100 particles with an average final goal error of approximately six percent. Conversely, we do have full transparency of the decision-making of the control system, while controlling all degrees of freedom of the source container with a mobile robot, and we also start every pouring run from a different position 50-100cm away from the destination container. Furthermore, this is not an inherent weakness of our framework, rather than a limitation of the employed symbolic theory. A theory adapted for precision pouring could include rules for a more precise flow control to achieve better results. For a fair comparison of that claim, we would have to integrate the same fill level measurement method as the related work in our future works.

Looking at the spilling rates, our system is able to adapt the pouring pose accordingly to avoid further spilling, but this does not include singular particles that occasionally spill, as they are not classified as spillage. Therefore, some amount of spillage must occur for our system to react to it. That the system is capable of reacting to spilling can be seen by the low amount of particles that are spilled compared to the number of all particles in the cup (140). Another mode of spilling occurs when the cup is already pouring without spilling, but the reasoner commands to move the cup to avoid potential spilling. As the velocity gains are fixed, the subsequent movement can be too fast, which causes spilling of some particles. Therefore, future work should explore reasoning about the velocity gain to react slower or faster in some situations, or a continuous output for velocity gains. The related works did not measure spillage rates.

An interesting observation happened when the cup was tilted to the left, the HSR reached a position limit in its wrist rotation joint, theoretically preventing it from tilting any further. The Full-Body controller solved this by using the combined movement of the arm and the base of the HSR to realize the full pouring movement. Highlighting the importance of full-body task space control for household robots in contrast to optimizing one specific degree of freedom with a PID controller.

Table 1: Outcomes of pouring different quantities of particles from various positions. Experiments were conducted with the HSR and two cups(see Figure 3). For each row, ten runs were performed; we present the average and standard deviation of particles exceeding the goal or being spilled.

Goal	Goal Error [#]	Spilling [#]	Sliding Friction Coefficient
10 particles	12.6 ± 2.46	1.4 ± 1.35	1
40 particles	13.1 ± 6.42	7.5 ± 7.67	1
100 particles	6.2 ± 3.49	9.3 ± 6.67	1
10 particles	10.3 ± 8.65	6.4 ± 10.04	3
40 particles	18.7 ± 8.08	6.3 ± 5.6	3
100 particles	5.5 ± 5.99	8.2 ± 6.86	3

6.3 Draining from a Pot

Draining is a variation of pouring in the sense that one substance is poured from a source container, while a second substance with different physical properties should stay within the source container. This is simulated by placing a larger ball in a pot with 100 other particles, as seen in Figure 4. The larger particle has a higher friction coefficient, so it is possible to separate both substances by pouring. Moreover, since the container in this instance is a cuboid, pouring from one of its corners offers greater control and precision compared to pouring along the edges. Based on this feature of the pot, a lower corner of the tilted pot is aligned toward the destination. This is encoded in the employed symbolic theory that extends the standard theory for pouring. A further extension is that whenever the pot is tilted, the position of the large particle, the retained substance, is monitored to tilt back whenever it is too close to the rim. To execute the draining task, the pot is grasped with both grippers of the bimanual PR2 and then the proposed system is initialized. It is parameterized to control the task frame in the center of the pot, and the reasoner is initialized with the action (draining), the relevant objects (seen in Figure 4), and the goal (pour 40 particles and have the large particle retained in the pot).

The effects of the adapted symbolic theory can be seen on the right side of Figure 4, where the conclusion to tilt back when the retained substance is close to the rim of the pot allows the reasoner to deactivate the fact that pouring is possible whenever the large particle is in danger of falling out of the pot. This in turn leads the reasoner to conclude the tiltBack motion primitive, as the pot should not be tilted when it is not possible to pour, which causes the large particle to move away from the edge of the pot, which in turn reactivates the fact that pouring is possible. This leads to a cycle that continues until enough particles are in the destination container.

Table 2: Outcomes of draining different quantities of particles from initially 100 particles. Experiments were conducted with the PR2 and two pots (see Figure 4). The goal describes the umber of particles that should be in the destination pot. For each row, ten runs were performed; we present the average and standard deviation of poured particles deviating from the goal and being spilled.

Goal	Goal Error [#]	Spilling [#]
10 particles	4.6 ± 4.7	0.3 ± 0.49
40 particles	3.9 ± 3.21	1.2 ± 1.14
70 particles	1.2 ± 1.48	2.5 ± 1.9
100 particles	-9.8 ± 3.77	5 ± 3.6

We also executed this experiment 40 times with different goal amounts of particles that should be drained from one pot to another. The results can be seen in Table 2. The data show, that in general the error is lower than that for pouring. This is due to the different opening of the pot, where fewer particles come out in bulk at once. Also, the trends of overshooting the goal and that the higher the goal is, the lower the error continues. An exception to this is the case where all 100 particles should be drained from the source pot into the destination pot. Here, the system always under performs; due to the particles that are spilled, they cannot be drained into the pot anymore, and due to a few particles that are held back by the larger ball and will not fall out. Therefore, we stopped draining after about 3 minutes in each run. When deducting the spilled particles from the goal error, on average 4.8 particles are left in the source container. We had to stop the draining controller manually, as this experiment discovered a limitation of our employed theory, where it did not consider that it could be impossible to completely empty a container. Once such a limitation is detected, the design of our framework allows us to extend the symbolic theory to deal with the limitation. In the case of the defeasible rule-based reasoner, we added a rule that negates that pouring is possible when a small amount of particles is left in the pot during draining. In this experiment, we set the small amount to be less than six particles. By assigning it a higher priority than the rule that makes pouring possible as long as the goal is not reached, we can successfully handle the discovered limitation.

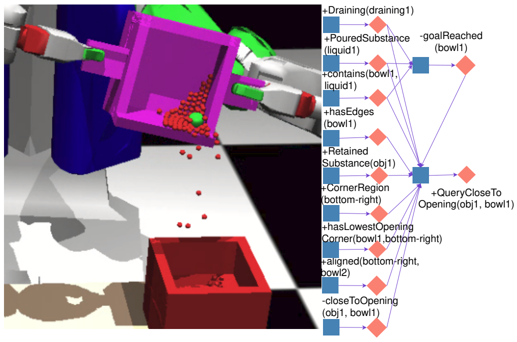

Figure 4: The PR2 draining a pot from a corner, and the reasoning graph inferring to observe the large green ball to keep it inside the pot.

6.4 Scraping from a Cutting Board

We consider the action of scraping sticky objects from a cutting board (Figure 5) as a variation of pouring because it achieves the same effect as tilting the cutting board to transfer something from it into a bowl when the tilting action alone is not enough. The sticky cubes on the cutting board are simulated using the adhesion feature of Mujoco. To execute the scraping task, the cutting board is grasped with one hand and the second hand is placed at the end of the cutting board. The system is then initialized with the action (to transfer), the objects (seen in Figure 5), and the goal (to transfer two cubes into the bowl). Additionally, this desired twist constraint for the second gripper is included in the controller specification:

$$\xi_{d2} = \begin{pmatrix} v_{d2} \\ \omega_{d2} \end{pmatrix} = \begin{pmatrix} ap_+ - ap_- \\ \mathbf{0} \end{pmatrix} \quad (3)$$

where $\mathbf{0} \in \mathbb{R}^5$ is a vector consisting of only zeroes. This constraint is added to the initial twist constraint that is initialized to act on a frame in the center of the cutting board. The new constraint moves the coordinate frame of the second gripper back and forth along

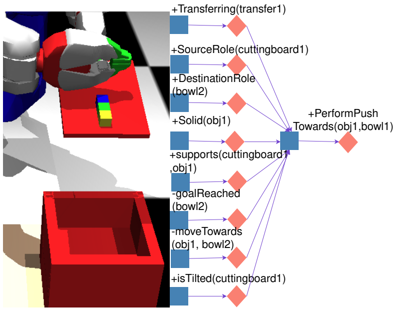

Figure 5: The PR2 scraping sticky cubes from a cutting board, and the reasoning graph inferring the pushing motion.

the cutting board. The symbols p_+, p_- correspond to new motion primitives called pushMore and pushLess that are added to the symbolic theory for the scraping action. The reasoning procedure is again an extension of the standard theory for pouring where motion primitives are added for the increased action space of the task. Figure 5 shows that the cutting board is already tilted but the objects do not move owing to their stickiness, therefore it is concluded that the objects should be pushed towards the bowl using the added motion primitives. This could even be extended to control all degrees of freedom of the the second gripper. But this example is already sufficient to showcase the flexibility and utility of the proposed knowlede-based servoing framework that is achieved by just converting the qualitative human understanding of the task into a tractable set of rules.

7 CONCLUSIONS

In this paper, we introduced knowledge-based servoing as a paradigm for embedding symbolic reasoning directly into a closed-loop control framework. Our evaluation in Section 6 focused on a set of pouring-related tasks (transferring liquids, draining mixtures, scraping sticky materials) and demonstrated the framework's flexibility across varied requirements, robots, and simulation setups. Despite some performance trade-offs compared to highly specialized controllers, the approach yielded transparent task execution and human-understandable failure modes, illustrating the value of symbolic theories in robotic control.

Beyond pouring tasks, the framework's ability to "plug in" different symbolic theories paves the way for broader applications in real-world household scenarios. The use of defeasible reasoning promotes straightforward debugging and adaptation, an important benefit for robots operating in unstructured environments or collaborating safely with humans. As a result, the methodology can help advance dependable and trustworthy manipulation solutions, bridging the gap between high-level cognitive reasoning and precise motion control.

Looking ahead, a key challenge lies in transitioning from simulation to hardware. Robust perception of semantic features (e.g., fill level or spill detection) and mitigating occlusions with camera-based input will require additional sensing modalities or advanced neuro-symbolic perception techniques (Pomarlan et al., 2024; De Giorgis et al., 2024), potentially leveraging large vision-language models. Moreover, real-world experiments must validate control frequency and stability to ensure safe deployment. Nonetheless, the demonstrated resilience of our motion controller across different robots provides a strong foundation for further exploration, including more complex tasks and domains.

In summary, knowledge-based servoing offers a path toward robotics systems that can be both versatile and interpretable. By coupling symbolic reasoning with real-time control, this framework highlights a promising avenue for enabling robots to adapt to new tasks, explain their decisions, and ultimately perform household manipulation in a manner that is both effective and transparent.

ACKNOWLEDGEMENTS

This work was supported by the German Research Foundation DFG, as part of Collaborative Research Center (Sonderforschungsbereich) 1320 Project-ID 329551904 "EASE - Everyday Activity Science and Engineering", University of Bremen (http://www.ease-crc.org/). The research was conducted in subprojects "R04 – Cognition-enabled execution of everyday actions" and "P01 – Embodied semantics for the language of action and change: Combining analysis, reasoning and simulation" This work was also supported by the European Union's Horizon 2020 research and innovation program under grant agreement No 101017089 as part of the TraceBot project.

REFERENCES

Aertbeliën, E. and De Schutter, J. (2014). etasl/etc: A constraint-based task specification language and robot controller using expression graphs. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1540–1546.

Antoniou, G., Billington, D., Governatori, G., Maher, M. J., and Rock, A. (2000). A family of defeasible reasoning logics and its implementation. In *Proceedings of the 14th European Conference on Artificial Intelligence*, ECAI'00, page 459–463, NLD. IOS Press.

Bouyarmane, K., Chappellet, K., Vaillant, J., and Kheddar, A. (2018). Quadratic programming for multirobot and task-space force control. *IEEE Transactions on Robotics*, 35(1):64–77.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., Florence, P., Fu, C., Arenas, M. G., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., Hsu, J., Ichter, B., Irpan, A., Joshi, N., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, L., Lee, T.-W. E., Levine, S., Lu, Y., Michalewski, H., Mordatch, I., Pertsch, K., Rao, K., Reymann, K., Ryoo, M., Salazar, G., Sanketi, P., Sermanet, P., Singh, J., Singh, A., Soricut, R., Tran, H., Vanhoucke, V., Vuong, Q., Wahid, A., Welker, S., Wohlhart, P., Wu, J., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., and Zitkovich, B. (2023). Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*.

Chaumette, F. and Hutchinson, S. (2006). Visual servo control. i. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90.

Corke, P. and Haviland, J. (2021). Not your grandmother's toolbox–the robotics toolbox reinvented for python. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11357–11363. IEEE.

De Giorgis, S., Pomarlan, M., and Tsiogkas, N. (2024). *ISD8 Tutorial Report: Cognitively Inspired Reasoning for Reactive Robotics-From Image Schemas to Knowledge Enrichment*.

Dong, C., Takizawa, M., Kudoh, S., and Suehiro, T. (2019). Precision pouring into unknown containers by service robots. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5875–5882.

Escande, A., Mansard, N., and Wieber, P.-B. (2014). Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028.

Ferretti, E., Errecalde, M., Garcia, A., and Simari, G. (2007). An application of defeasible logic programming to decision making in a robotic environment. In *Logic Programming and Nonmonotonic Reasoning (LPNMR)*, pages 297–302.

Gibson, J. J. (1977). The theory of affordances. In Robert E Shaw, J. B., editor, *Perceiving, acting, and knowing: toward an ecological psychology*, pages pp.67–82. Hillsdale, N.J. : Lawrence Erlbaum Associates.

Guo, H., Wu, F., Qin, Y., Li, R., Li, K., and Li, K. (2023). Recent trends in task and motion planning for robotics: A survey. *ACM Comput. Surv.*, 55(13s).

Johnson, M. (1987). *The body in the mind: The bodily basis of meaning, imagination, and reason*. The body in the mind: The bodily basis of meaning, imagination, and reason. University of Chicago Press, Chicago, IL, US.

Kansky, K., Silver, T., Mély, D. A., Eldawy, M., Lázaro-Gredilla, M., Lou, X., Dorfman, N., Sidor, S., Phoenix, S., and George, D. (2017). Schema networks: Zero-shot transfer with a generative causal model of intuitive physics.

Kress-Gazit, H., Wongpiromsarn, T., and Topcu, U. (2011). Correct, reactive, high-level robot control. *Robotics & Automation Magazine, IEEE*, 18:65 – 74.

Lam, H.-P. and Governatori, G. (2013). Towards a model of uavs navigation in urban canyon through defeasible logic. *J. Log. and Comput.*, 23(2):373–395.

Lindemann, L. and Dimarogonas, D. V. (2019). Control barrier functions for signal temporal logic tasks. *IEEE Control Systems Letters*, 3(1):96–101.

Mandler, J. M. (1992). How to build a baby: Ii. conceptual primitives. *Psychological review*, 99(4):587.

Mansard, N., Stasse, O., Evrard, P., and Kheddar, A. (2009). A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks. In *International Conference on Advanced Robotics (ICAR)*, page 119.

Meli, D., Nakawala, H., and Fiorini, P. (2023). Logic programming for deliberative robotic task planning. *Artificial Intelligence Review*, 56.

Muhayyuddin, Akbari, A., and Rosell, J. (2017). Physics-based motion planning with temporal logic specifications. *IFAC-PapersOnLine*, 50(1):8993–8999. 20th IFAC World Congress.

Pan, Z., Park, C., and Manocha, D. (2016). Robot motion planning for pouring liquids. *Proceedings of the International Conference on Automated Planning and Scheduling*, 26(1):518–526.

Piacenza, P., Lee, D., and Isler, V. (2022). Pouring by feel: An analysis of tactile and proprioceptive sensing for accurate pouring. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10248–10254.

Pomarlan, M., De Giorgis, S., Ringe, R., Hedblom, M. M., and Tsiogkas, N. (2024). Hanging around : Cognitive inspired reasoning for reactive robotics. In *Formal Ontology in Information Systems : Proceedings of the 14th International Conference (FOIS 2024)*, number 394 in Frontiers in Artificial Intelligence and Applications, pages 2–15.

Schenck, C. and Fox, D. (2017). Visual closed-loop control for pouring liquids. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2629–2636.

Shanahan, M. and Witkowski, M. (2001). High-level robot control through logic. In Castelfranchi, C. and Lespérance, Y., editors, *Intelligent Agents VII Agent Theories Architectures and Languages*, pages 104–121, Berlin, Heidelberg. Springer Berlin Heidelberg.

Stelter, S., Bartels, G., and Beetz, M. (2022a). An open-source motion planning framework for mobile manipulators using constraint-based task space control with linear mpc. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1671–1678. IEEE.

Stelter, S., Bartels, G., and Beetz, M. (2022b). An open-source motion planning framework for mobile manipulators using constraint-based task space control with linear mpc. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1671–1678. IEEE.

Xiao, W., Mehdipour, N., Collin, A., Bin-Nun, A. Y., Frazzoli, E., Tebbens, R. D., and Belta, C. (2021). Rule-based optimal control for autonomous driving. In *Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems*, ICCPS '21, page 143–154, New York, NY, USA. Association for Computing Machinery.

Zhu, F., Hu, S., Letian, L., Bartsch, A., George, A., and Farimani, A. B. (2023). Pour me a drink: Robotic precision pouring carbonated beverages into transparent containers. *arXiv preprint arXiv:2309.08892v2*.