2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Composable Causality in Semantic Robot Programming

Emily Sheetz, Xiaotong Chen, Zhen Zeng, Kaizhi Zheng, Qiuyu Shi, and Odest Chadwicke Jenkins

***Abstract*— Assembly tasks are challenging for robot manipulation because the robot must reason over the composed effects of actions and execute multi-objective behaviors. Robots typically use predefined priorities provided by users to determine how to compose controller behaviors, but we want the robot to autonomously select these compositions based on their composed effects within the task. We present Composable Causality in Semantic Robot Programming to allow robots to reason over the composed effects of controllers when executing multi-objective actions and autonomously compose controllers without predefined priorities. Our proposed *causal control basis* combines controller behaviors with causal information about how the behaviors can be used to execute high-level symbolic actions. The robot uses the *causal control basis* to predict the transition probability of achieving the composed effects of a multi-objective action. The composed causality estimates are used to select which action to execute within the context of a furniture assembly task. We evaluate the robot's transition probability estimates in different furniture assembly trials in simulation on the Baxter robot. The robot's ability to assemble furniture using different multi-objective connection actions demonstrates the usefulness of the composed causality estimates from our *causal control basis*.**

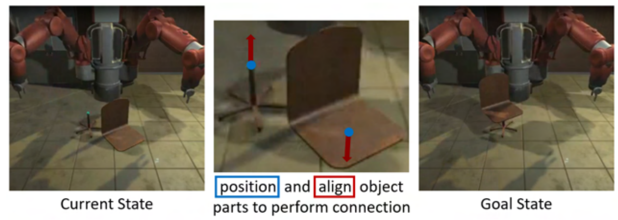


Fig. 1: A robot assembling furniture through goal-directed manipulation. The robot must reason over the composable causality of controllers to achieve the task goal.

I. INTRODUCTION

Assembly tasks present unique challenges in reasoning over objects and executing complex behaviors in long-horizon tasks. Robots have difficulty assembling objects because they have to compose the effects of multiple behaviors and maintain these composed effects as they move on to the next step. We need a flexible way to program our robots to perform assembly tasks and Semantic Robot Programming (SRP) [24] has emerged as an intuitive way to declaratively program robots. Within the SRP paradigm, robots can infer goal conditions from a demonstrated goal scene and reason over available objects and actions to reach the goal. The focus of SRP was to address the perceptual challenges involved in perceiving the demonstrated goal conditions. Now that these perceptual challenges have been addressed, we aim to extend SRP in terms of the types of actions that the robot can perform. By extending SRP, we will be able to declaratively program robots to perform actions with composed effects on objects, as is expected in challenging assembly tasks. Specifically, robots need to reason about the objects themselves, compose effects on these objects during task execution, and overcome the challenges of assembly tasks.

To execute actions, it is common to use off-the-shelf motion planners. While motion planners allow robots to plan how to move from one point to another, they do not provide the capabilities necessary to perform complex manipulation tasks, especially assembly tasks. Motion planners do not allow robots to move from one point to another while achieving concurrent motion goals—for example, screwing in a screw involves moving and performing a spiraling motion concurrently—or react quickly to changes in the environment. To move beyond the capabilities of motion planners, many works use *object-centric controllers* within a *control basis*. Controllers can also be composed together to execute more complex *multi-objective actions*, meaning multiple controllers are executed concurrently. Controllers within a control basis provide a more robust, reactive, and capable approach to executing the complex, multi-objective actions expected in assembly tasks.

The compositions of controller behaviors necessary to perform multi-objective actions are generally determined by a predefined priority provided by the user. However, we want robots to autonomously compose controllers without these predefined priorities. Reasoning over actions symbolically can be disconnected from the realities of physical execution. Instead, we expect robots to reason over the preconditions and postconditions of the executable action and ground the action effects on the objects in the perceived scene. Only recently have researchers started to address the question of autonomously composing controllers for multi-objective actions [21]. Figure 1 shows an example of the challenges a robot faces while executing a multi-objective action during an assembly task. It is difficult to express the qualitative insights users have on controller compositions in a precise, quantitative way that can be reasoned over and used by robots. Instead, we want to place the responsibility on the robot to discover why one controller composition performs better in some situations than others, lessening the cognitive load on the user. The robot will need to reason over the composed effects of different compositions autonomously in order to execute multi-objective actions within the SRP

E. Sheetz, X. Chen, Z. Zeng, K. Zheng, Q. Shi, and O. C. Jenkins are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA [esheetz | cxt | zengzhen | zhengkz | shiqy |ocj]@umich.edu

paradigm.

In this paper, we propose a *causal control basis* to build on SRP and allow robots to autonomously compose controllers to achieve assembly tasks. We take inspiration from works that apply notions of *causality* to goal-directed manipulation tasks [23]. Using the given *causal control basis*, the robot can estimate the transition probability that a controller composition will achieve the desired composed effects on the objects in the scene. During task execution, the robot will autonomously compose controller behaviors based on their predicted composed effects and execute the afforded multi-objective actions to assemble a piece of furniture. We test our *causal control basis* in simulation on the Baxter robot in a variety of furniture assembly tasks and find that it provides the robot with sufficient information to autonomously compose controllers without predefined priorities from the user. Our work on **Composable Causality in Semantic Robot Programming** demonstrates that the *causal control basis* allows the robot to achieve challenging goal-directed manipulation tasks within the SRP paradigm.

II. RELATED WORK

A. Assembly Tasks

We consider construction tasks as an interesting domain for complex goal-directed manipulation. Assembly is a challenging problem because it requires the ability to plan over long-horizons, understand properties of objects, and manipulate objects in particular ways. Nair *et al.* [10] present a tool construction pipeline that allows robots to construct tools to achieve tasks. Many works use learning from demonstration to teach reusable motion primitives that the robot uses to assemble different objects [25], [22], including furniture such as tables [11]. Lee *et al.* [8] developed the IKEA Furniture Assembly Environment as a test-bed for the perception, planning, and control required to perform construction tasks. Though their assembly environment is designed for reinforcement learning, we use the IKEA Furniture Assembly Environment to allow the robot to predict the effects of a given control policy. Together, these works emphasize the immense interest and challenge of tool-use and construction tasks, which motivates our choice of assembly tasks as our problem domain.

B. Object-Centric Controllers and Control Basis

Due to the significance of object affordances [5], robotics research has gravitated towards interacting with objects through *object-centric* motions. Object-centric behaviors are expressed in *task frames* that bridge the gap between high-level symbolic description of actions and the low-level servomechanism execution of actions [1], [6]. For example, expressing a crank-turning-action in world frame involves reasoning over the arc the crank might follow. But expressing the same crank-turning-action in a task frame fixed to the crank involves applying force along an axis of the crank until resistance is met [1]. Reasoning in the task frame simplifies the expression of the action and emphasizes the effect on the object itself.

To execute these object-centric actions, researchers use object-centric controllers, which send joint commands such that the robot achieves low-level motion primitives. Controllers offer advantages over motion planners because they can be used within a *control basis* that forms the building blocks of all behaviors the robot might need to execute. The behaviors within a control basis can be *composed* to yield multi-objective behaviors, meaning multiple controllers are being executed concurrently [19]. For example, robust grasping could be formulated as a multi-objective action that involves *positioning* an end-effector while *aligning* the approach axis of that end-effector with the target object. Composing multiple behaviors induces a priority between these behaviors, which means one behavior will likely be achieved first and in the worst case may impede the other controller(s) from converging. The priorities between controllers—the particular composition of these controllers—can greatly impact the effect of the multi-objective action. Many works explore composing controllers in atomic actions such as grasping [16], [17], [18] or conditioning behaviors [7] such as avoiding joint limits and singularities.

Compositions of controller behaviors are generally determined by a predefined priority provided by the user. For example, a user may determine experimentally that prioritizing positioning over alignment results in the most robust grasp poses. Therefore, the user will hard-code the robot to always perform multi-objective grasps by composing these behaviors such that positioning is the highest priority. However, we want robots to autonomously compose controllers to increase their reasoning power over complex manipulation actions. Sharma *et al.* [21] present a reinforcement learning approach to determining how controllers should be composed to perform different tasks. They demonstrate that their approach allows robots to autonomously compose controllers in atomic actions such as block pushing, screw turning, and door opening. The work of Sharma *et al.* demonstrates a significant step towards autonomously composing controllers in order to perform atomic actions. We build on this work and extend it to long-horizon construction tasks, where it is necessary to reason about how the multi-objective action will be used in sequence to achieve a larger task goal. This distinction requires the robot to have a deeper understanding of when controller behaviors can be enacted, what the composed effects of multi-objective actions will be, and how to maintain these composed effects throughout task execution.

C. Causality

Work on causality analyzes cause and effect relationships between variables. Causal relationships can be expressed as Causal Bayesian networks (CBNs) [4] and analyzed through queries [13], [14], [15]. Traditionally, work on *causality* refers to the effects of changes to variables on the distributions of other random variables in the system. In the context of robot manipulation, *causality* takes on a different meaning, referring to how robot actions cause effects on objects in the scene. Causality has been particularly helpful in allowing robots to reason over long-horizon tasks, such as assembly tasks. Work in robot manipulation often uses several causal models. Xiong *et al.* [23] found that hierarchical spatial, temporal, and causal models can be learned from demonstration and used to achieve cloth-folding tasks. We build on this work by incorporating hierarchical temporal and causal models into our proposed *causal control basis* to allow robots to autonomously compose controllers and reason over the composed effects of actions in long-horizon tasks.

III. METHODS

A. Problem Formulation

To perform assembly tasks that require multi-objective behaviors, the robot needs to predict the *transition function* of the controllers—the probability that a given composition of controllers will achieve their composed effects. We assume we have a control basis Φ of controllers that can be composed using nullspace projection to achieve multiple objectives. Given a task goal, the robot constructs a high-level task plan using an off-the-shelf task planner and decomposes each symbolic action into a sequence of executable motions. For actions that require multi-objective behaviors, we want the robot to autonomously compose the given controllers and execute the planned symbolic action by reasoning over the *causality* of the controllers based on the transition probability estimates.

We formulate this probabilistic planning problem as a Markov Decision Process (MDP) [9], [20] (S, A, P, C). The state space S is determined by the robot configuration space and the poses of the objects in the scene. A control basis is a set of controllers $\Phi = \{\phi_i\}_{i=0}^{N}$. The action space A is the set of all possible controllers and compositions in the given control basis Φ. The controllers that can be running at any given time are elements of the power set of the control basis $\mathcal{P}(\Phi)$. Suppose we have controllers ϕ_i and ϕ_j that achieve objectives i and j, respectively. One possible composition of these controllers is $\phi_j \lhd \phi_i$, where the "subject-to" relation $\lhd$ indicates that ϕ_i has a higher priority than ϕ_j. Let $M_t \in \mathcal{P}(\Phi)$ be the set of controllers running at time t. Since composing controllers induces an ordering (priority) between them, all possible compositions of the running controllers M_t are elements of the symmetric group S_{M_t}, which is the set of all possible permutations over the elements (controllers) in M_t. For example, if $M_t = \{\phi_i, \phi_j\}$, then $S_{M_t} = \{\phi_i \lhd \phi_j, \phi_j \lhd \phi_i\}$. Therefore, the action space for control basis Φ is $A = \{S_{M_t} \mid M_t \in \mathcal{P}(\Phi)\}$. The transition probability $P(s' \mid s, a)$ indicates the probability of achieving the composed effects s' of a (composed) controller $a \in A$ when enacted in the current state s. The cost function $C_a(s)$ is the cost of enacting controller a in state s. We want the robot to execute the controller a that will achieve its composed effects by minimizing its objective function ϕ, meaning $C_a(s) = \phi_a(s)$.

The action space is determined by the causal control basis, but the robot is not given any information about the transition probabilities associated with the (composed) controllers. The robot needs to estimate the transition probability for each possible controller composition.

B. Causal Control Basis

We propose a *causal control basis* $\mathbf{\Phi}$ that the robot will use to predict the transition probabilities of actions and determine which composition of controllers to execute to achieve assembly tasks. The *causal control basis* is given to the robot and is comprised of the following components:

- The implemented controllers in the control basis Φ, which can be combined using *nullspace composition* to perform multi-objective actions and enact the *affordances* in the scene.
- The set of temporal graphs G_T, which represent the sequence of controllers that correspond to high-level symbolic actions. The root of the temporal graph is the high-level symbolic action, and each low-level controller behavior required to execute this action is a child of the root, arranged from left-to-right in sequence. The structure of the temporal graph is inspired by previous work that uses hierarchical graphs within robot manipulation tasks [23].
- The set of composed causal graphs G_C, which describe what controllers are involved in a multi-objective action. Causal graphs are comprised of *preconditions*, which are literals that must be true before an action is taken; *controller behaviors* that must be executed concurrently to achieve a multi-objective action; the *effects* of the individual controllers; and the desired *composed effects*, which are literals that should be achieved by the composition of the controllers.

Our *causal control basis* is denoted $\mathbf{\Phi} = (\Phi, G_T, G_C)$. For the tasks considered in the experiments, we define the control basis—discussed in more detail in Section IV-A.1—to include pose, position, rotation, and screw controllers. However, our formulation will work with an arbitrary control basis.

To predict the transition probabilities of the controller compositions, the *causal control basis* performs *offline walkouts*. Before task execution, the robot uniformly samples initial states $s \in S$ and controller goals $s' \in S$. Suppose the robot is predicting the transition probability for arbitrary controller composition $\phi_k \lhd \phi_j \lhd \phi_i$ (where the "subject-to" relation $\lhd$ indicates the priority of behaviors in the composition). The robot simulates execution of the composed controllers until they converge or until large time threshold T. The predicted transition probability that the action $a = \phi_k \lhd \phi_j \lhd \phi_i$ achieves the composed effects s' based on the offline walkout is:

$$\hat{P}(s' \mid s, a) = \begin{cases} 1 & \text{objectives met} \\ 0 & \text{bad progress} \\ \frac{\phi_a(s) - \phi_a(s_T)}{\phi_a(s)} & \text{otherwise} \end{cases} \quad (1)$$

where s_T is the state at time threshold T and ϕ_a indicates the composed cost or objective function values at the given state. When the controllers meet their objectives, $\phi_a(s') = 0$. If bad progress is made, then at some state s_t, the controllers reached a local minimum—objective function $\phi_a(s_t)$

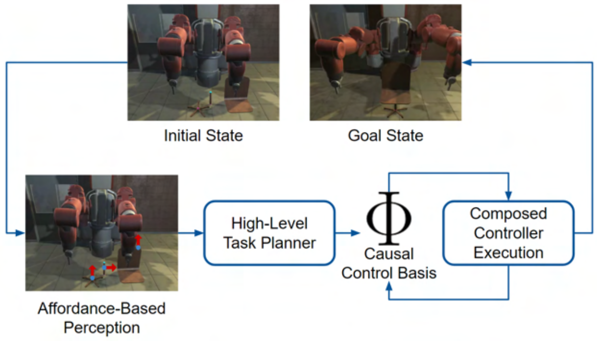


Fig. 2: Our pipeline for Composable Causality in Semantic Robot Programming.

stopped decreasing—or the objective function has increased beyond the starting value—$\phi_a(s_t) > \phi_a(s)$. Otherwise, the controllers made some progress to decreasing the objective function. For a large number of random samples, the average predicted transition probability indicates how the controller composition performs across action instances. During task execution, the robot will query the *causal control basis* for the controller composition with the greatest predicted transition probability and will execute that composition.

IV. EXPERIMENTS AND RESULTS

Figure 2 describes the pipeline for assembling furniture using Composable Causality in Semantic Robot Programming and the use of our proposed *causal control basis*. We assume that the robot has parsed the goal conditions from a demonstrated goal scene of the task as in SRP [24] and that we have affordance-based perception[1] to perceive the objects and affordances in the scene. These perceived objects and affordances seed the initial state of an off-the-shelf high-level task planner[2], which constructs the task plan. The *causal control basis* converts each action in the high-level task plan into a sequence of (possibly composed) controller commands and instantiates the action based on the current poses of the objects and their connection sites. The robot executes this sequence of controllers to achieve the task goal of assembling furniture.

We evaluate the proposed *causal control basis* in various furniture assembly tasks in simulation using the Baxter robot in the IKEA Furniture Assembly Environment[3] [8]. We assume known object poses during manipulation and grasp poses for every object part are provided. The connection of two parts is implemented as welding in the Mujoco simulation, which checks the position and axis alignment of connecting points.

A. Causal Control Basis for Furniture Assembly

1) Control Basis Implementation: In this work, we define the control basis Φ for furniture assembly by 6D pose ϕ_{6Dpose}, 3D position ϕ_{pos}, rotation ϕ_{rot}, and screw ϕ_{screw} controllers. All of these controllers are object-centric potential field controllers based on attractive potential fields that attract the robot and objects to the controller goal. Our furniture assembly control basis Φ is the set of these controllers:

$$\Phi = \{\phi_{\text{6Dpose}}, \phi_{\text{pos}}, \phi_{\text{rot}}, \phi_{\text{screw}}\} \quad (2)$$

2) Temporal Graphs: The set of temporal graphs G_T indicate the sequence of controllers that correspond to the high-level `pick-up`, `insert`, and `screw` actions, as seen in Figure 3. For multi-objective connection actions `insert` or `screw`, the robot will have to determine what composition of the controllers (indicated in the corresponding causal graph in Figure 4) to execute within the sequence (indicated by the blue nodes in Figure 3) by selecting the composed controller with the maximum predicted transition probability.

3) Causal Graphs: The set of causal graphs G_C indicate the composed effects of the controllers within the multi-objective `insert` and `screw` actions. As shown in Figure 4, the causal graphs indicate the controllers that are involved in these connect actions, the pre-conditions of enacting these compositions, and the intended effects of these compositions. The robot will use the transition probability predictions from the *causal control basis* to determine how to compose these controllers together.

[1]For example, Affordance Coordinate Frames (ACFs) [2].
[2]Pyperplan STRIPS planning library: `https://github.com/aibasel/pyperplan`
[3]`https://clvrai.github.io/furniture/`

Composition a	Predicted Transition Probability $\hat{P}(s' \mid s, a)$
$\phi_{\text{pos}} \lhd \phi_{\text{rot}}$	**0.723**
$\phi_{\text{rot}} \lhd \phi_{\text{pos}}$	0.711

TABLE I: Transition probability predictions for `insert` action, based on 500 offline walkouts for each composition.

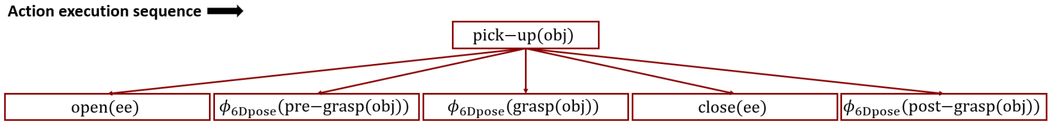
Composition a	Predicted Transition Probability $\hat{P}(s' \mid s, a)$
$\phi_{\text{rot}} \lhd \phi_{\text{screw}} \lhd \phi_{\text{pos}}$	**0.937**
$\phi_{\text{pos}} \lhd \phi_{\text{screw}} \lhd \phi_{\text{rot}}$	0.936
$\phi_{\text{screw}} \lhd \phi_{\text{pos}} \lhd \phi_{\text{rot}}$	0.929
$\phi_{\text{pos}} \lhd \phi_{\text{rot}} \lhd \phi_{\text{screw}}$	0.925
$\phi_{\text{screw}} \lhd \phi_{\text{rot}} \lhd \phi_{\text{pos}}$	0.923
$\phi_{\text{rot}} \lhd \phi_{\text{pos}} \lhd \phi_{\text{screw}}$	0.904

TABLE II: Transition probability predictions for `screw` action, based on 500 offline walkouts for each composition.

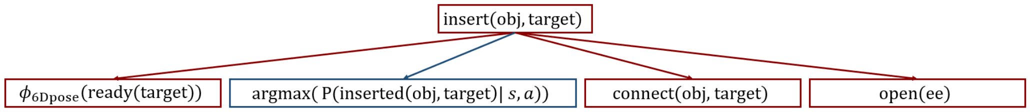
B. Composed Causality Predictions

For the `insert` and `screw` connection actions, the robot simulated 500 executions of each possible composition. We used time threshold $T = 300$ controller updates as the cutoff for the offline walkouts. The predicted transition probability $\hat{P}$ was computed as described in Section III-B.

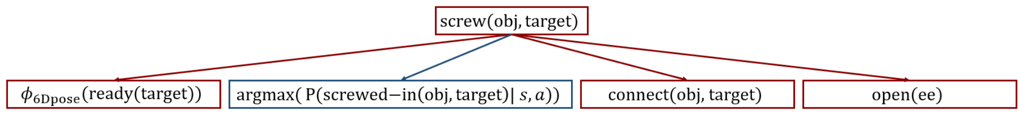
The transition probability predictions for the `insert` and `screw` actions are shown in Table I and Table II, respectively. The composition with the highest probability for the `insert` action indicates that positioning the object should be performed subject to aligning the object with the target, $\phi_{\text{pos}} \lhd \phi_{\text{rot}}$. The composition with the highest probability for the `screw` action indicates that aligning



(a) Temporal graph for `pick-up` action.

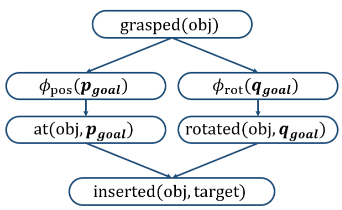


(b) Temporal graph for `insert` action.

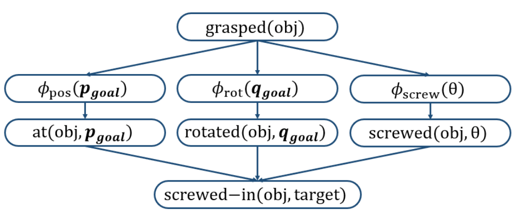


(c) Temporal graph for `screw` action.

Fig. 3: The temporal graphs relating high-level symbolic actions to the low-level controller behaviors in the control basis in sequence from left to right.



(a) Causal graph for `insert` action.



(b) Causal graph for `screw` action.

Fig. 4: The causal graphs for the multi-objective connect actions in the control basis for furniture assembly tasks.

the object should be performed subject to screwing and positioning the object, $\phi_{\text{rot}} \lhd \phi_{\text{screw}} \lhd \phi_{\text{pos}}$.

C. Furniture Assembly Task Results

The composition with the maximum transition probability prediction is used to execute connect actions in a variety of furniture assembly tasks to test the accuracy of the estimated composed causality of the controllers. We tested the `insert` action within 10 random trials of swivel chair assembly and tested the `screw` action within 10 random trials of table assembly. Across all trials, we compute the average task time and success rates of the `pick-up` action, the multi-objective `insert` and `screw` actions, and the entire assembly task. Summary information for the swivel chair trials are in Table III and table trials are in Table IV.

`Pick-Up` Action Success Rate	0.606
`Insert` Action Success Rate	**0.714**
Swivel Chair Assembly Task Success Rate	1
Average Execution Time (s)	266.241

TABLE III: Results from 10 swivel chair assembly tasks, with 33 `pick-up` attempts and 28 `insert` action attempts.

`Pick-Up` Action Success Rate	0.909
`Screw` Action Success Rate	**0.952**
Table Assembly Task Success Rate	1
Average Execution Time (s)	492.072

TABLE IV: Results from 10 table assembly tasks, with 44 `pick-up` attempts and 42 `screw` action attempts.

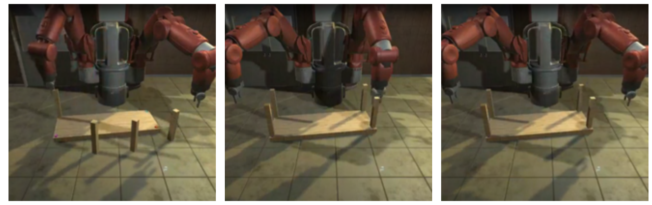
When actions failed, it was often due to joint limits or local minima being reached, especially during the `pick-up` actions. This is due to the predefined grasp poses that we assume are provided to the robot. Selecting grasp poses is not the responsibility of our *causal control basis*; instead, we assume we have known grasp poses, similar to affordance templates [6]. The success rates of the multi-objective connection actions reflect the performance of the *causal control basis*. For 28 `insert` attempts across 10 swivel chair assembly trials, the `insert` action success rate was 0.714 and the task success rate was 1. For 42 `screw` attempts across 10 table assembly trials, the `screw` action success rate was 0.923 and the task success rate was 1. The success rates for both actions in the swivel chair tasks are lower because the object parts required the robot to reach its arm much closer to the floor, and towards the limits of its reachable workspace. Images from three of the random swivel chair and table trials are in Figure 5.

The similarity of the multi-objective connection action success rates (Table III and Table IV) and the predicted transition probabilities (Table I and Table II) indicate that the transition probability predictions accurately capture the per-

(a) Swivel Chair Assembly Trail 1.

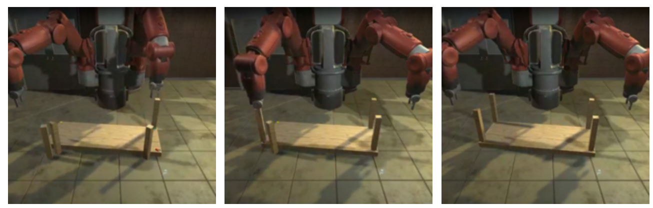
(b) Swivel Chair Assembly Trail 2.

(c) Swivel Chair Assembly Trail 3.



(d) Table Assembly Trial 1.

(e) Table Assembly Trial 2.



(f) Table Assembly Trial 3.

Fig. 5: Execution of three swivel chair assembly trials (left) and three table assembly trials (right) using multi-objective `insert` and `screw` actions to connect parts together, respectively.

formance of the compositions during task execution. When the connect actions did not result in successful connections (due to joint limits or collisions between objects), the robot would retry the action. The task success rate for both the swivel chair and table tasks indicate that the robot was able to recover in these cases and achieve a successful connection.

V. Discussion and Conclusion

The robot's ability to successfully assemble different furniture pieces demonstrates the accuracy of our proposed *causal control basis* in predicting the composed effects of controller behaviors. The *causal control basis* effectively extends the principles of Semantic Robot Programming—that we need an intuitive way to declaratively program robots to perform tasks—into more challenging tasks that involve multi-objective actions. The *causal control basis* describes the intended results of the multi-objective actions, but leaves the robot to determine how to compose the appropriate controller behaviors and perform the action successfully. The *causal control basis* represents important information that users often already provide robots in some form. For example, the causal graphs encode information similar to symbolic descriptions of action preconditions and postconditions and the temporal graphs encode information similar to hard-coded behavior sequences. Therefore, our *causal control basis* allows users to intuitively share knowledge about actions such that robots can autonomously determine how to enact those behaviors in challenging tasks.

Future work includes assembling more furniture pieces, assembling furniture with more challenging initial part poses, and implementing the controllers for a real-world robot rather than in simulation. Implementing our work on a real-world robot would also involve making the connection actions more realistic by considering depth of insertion and screwing. We will also create a more robust control basis by incorporating dynamics, including controller behaviors for avoiding joint limits and collisions between objects, and extending our *causal control basis* to coordinate the arms for bimanual manipulation tasks.

In this work, we proposed a *causal control basis* for achieving Composable Causality in Semantic Robot Programming. Our *causal control basis* allows the robot to predict the transition probabilities of controller compositions, thereby estimating the composed causality of multi-objective actions. Our work in Composable Causality in Semantic Robot Programming demonstrates that reasoning over a *causal control basis* provides the robot with the declarative knowledge necessary to autonomously compose controller behaviors without predefined priorities to achieve furniture assembly tasks.

Acknowledgments

This work was supported in part by NASA Space Technology Graduate Research Opportunity (NSTGRO) grant 80NSSC20K1200.

REFERENCES

[1] D. H. Ballard, "Task Frames in Robot Manipulation," *AAAI*, pp. 16-22, 1984.

[2] X. Chen, K. Zheng, Z. Zeng, S. Basu, J. Cooney, J. Pavlasek, and O. C. Jenkins, "Manipulation-Oriented Object Perception in Clutter through Affordance Coordinate Frames," *arXiv preprint arXiv:2010.08202*, 2020.

[3] F.-J. Chu, R. Xu, C. Tang, and P. A. Vela, "Recognizing Object Affordances to Support Scene Reasoning for Manipulation Tasks," *arXiv preprint arXiv:1901.08162*, 2019.

[4] I. Dasgupta, J. Wang, S. Chiappa, J. Mitrovic, P. Ortega, D. Raposo, E. Hughes, P. Battaglia, M. Botvinick, and Z. Kurth-Nelson, "Causal Reasoning from Meta-Reinforcement Learning," *arXiv preprint arXiv:1901.08162*, 2019.

[5] J. J. Gibson, "The Theory of Affordances," pp. 67-82, 1977.

[6] S. Hart, S. Dinh, and K. Hambuchen, "The Affordance Template ROS Package for Robot Task Programming," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6227-6234, 2015.

[7] S. Hart and R. Grupen, "Natural Task Decomposition with Intrinsic Potential Fields," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 2507–2512, 2007.

[8] Y. Lee, E. S. Hu, Z. Yang, A. Yin, and J. J. Lim, "IKEA Furniture Assembly Environment for Long-Horizon Complex Manipulation Tasks," *arXiv preprint arXiv:1911.07246*, 2019.

[9] M. L. Litman, T. L. Dean, and L. P. Kaelbling, "On the Complexity of Solving Markov Decision Problems," *arXiv preprint arXiv:1302.4971*, 2013.

[10] L. Nair, J. Balloch, and S. Chernova, "Tool Macgyvering: Tool Construction Using Geometric Reasoning," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5837–5843, 2019.

[11] S. Niekum, S. Chitta, A. G. Barto, B. Marthi, and S. Osentoski, "Incremental Semantically Grounded Learning from Demonstration," *Robotics: Science and Systems (RSS)*, 2013.

[12] J. Pavlasek, S. Lewis, K. Desingh, and O. C. Jenkins, "Parts-Based Articulated Object Localization in Clutter Using Belief Propagation," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[13] J. Pearl, "Causal Diagrams for Empirical Research," *Biometrika*, vol. 82, no. 4, pp. 669–688, 1995.

[14] J. Pearl, *Causality*. Cambridge University Press, 2009.

[15] J. Pearl, "The Do-Calculus Revisited," *arXiv preprint arXiv:1210.4852*, 2012.

[16] R. Platt Jr, A. H. Fagg, and R. A. Grupen, "Manipulation Gaits: Sequences of Grasp Control Tasks," *IEEE International Conference on Robotics and Automation (ICRA)* vol. 1, pp. 801–806, 2004.

[17] R. Platt Jr, A. H. Fagg, and R. A. Grupen, "Nullspace Composition of Control Laws for Grasping," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[18] R. Platt Jr, A. H. Fagg, and R. A. Grupen, "Null-Space Grasp Control: Theory and Experiments," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 282–295, 2010.

[19] K. Rohanimanesh, R. Platt, S. Mahadevan, and R. Grupen, "Coarticulation in Markov Decision Processes," *Advances in Neural Information Processing Systems*, 2004.

[20] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, Third Edition*. Pearson Education, 2010.

[21] M. Sharma, J. Liang, J. Zhao, A. LaGrassa, and O. Kroemer, "Learning to Compose Hierarchical Object-Centric Controllers for Robotic Manipulation," *arXiv preprint arXiv:2011.04627*, 2020.

[22] Y. Wang, R. Xiong, L. Shen, K. Sun, J. Zhang, and L. Qi, "Towards Learning from Demonstration System for Parts Assembly: A Graph Based Representation for Knowledge," *IEEE International Conference on Cyber Technology in Automation, Control and Intelligent*, pp. 174-179, 2014.

[23] C. Xiong, N. Shukla, W. Xiong, and S.-C. Zhu, "Robot Learning with a Spatial, Temporal, and Causal And-Or Graph," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2144–2151, 2016.

[24] Z. Zeng, Z. Zhou, Z. Sui, and O. C. Jenkins, "Semantic Robot Programming for Goal-Directed Manipulation in Cluttered Scenes," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7462–7469, 2018.

[25] Z. Zhu and H. Hu, "Robot Learning from Demonstration in Robotic Assembly: A Survey," *Robotics*, 2018.