Supplementary Document

Defining action parameters for the nine common atomic actions – an explanation

Authors: Hao Zheng, Regina Lee

Supervisor: Yuqian Lu

Date: 26/4/2023

Version: 1.0.0

This is the supplementary document for human-robot shared assembly taxonomy (HR-SAT). More details of the taxonomy can be found at: https://iai-hrc.github.io/hr-sat.

In this document, we explain how to define action parameters for nine common atomic actions (see Fig. 1) defined in our paper. The action parameters are summarized in Table 1. When we define the action parameters, we follow three principles: (a) provide action parameters that a complete description of the atomic action required for both human and robot; (b) provide action parameters that common robot programming methods need; (c) consider the needs of various applications and provide comprehensive action parameters as much as possible; and (d) avoid repetitive parameter definitions.

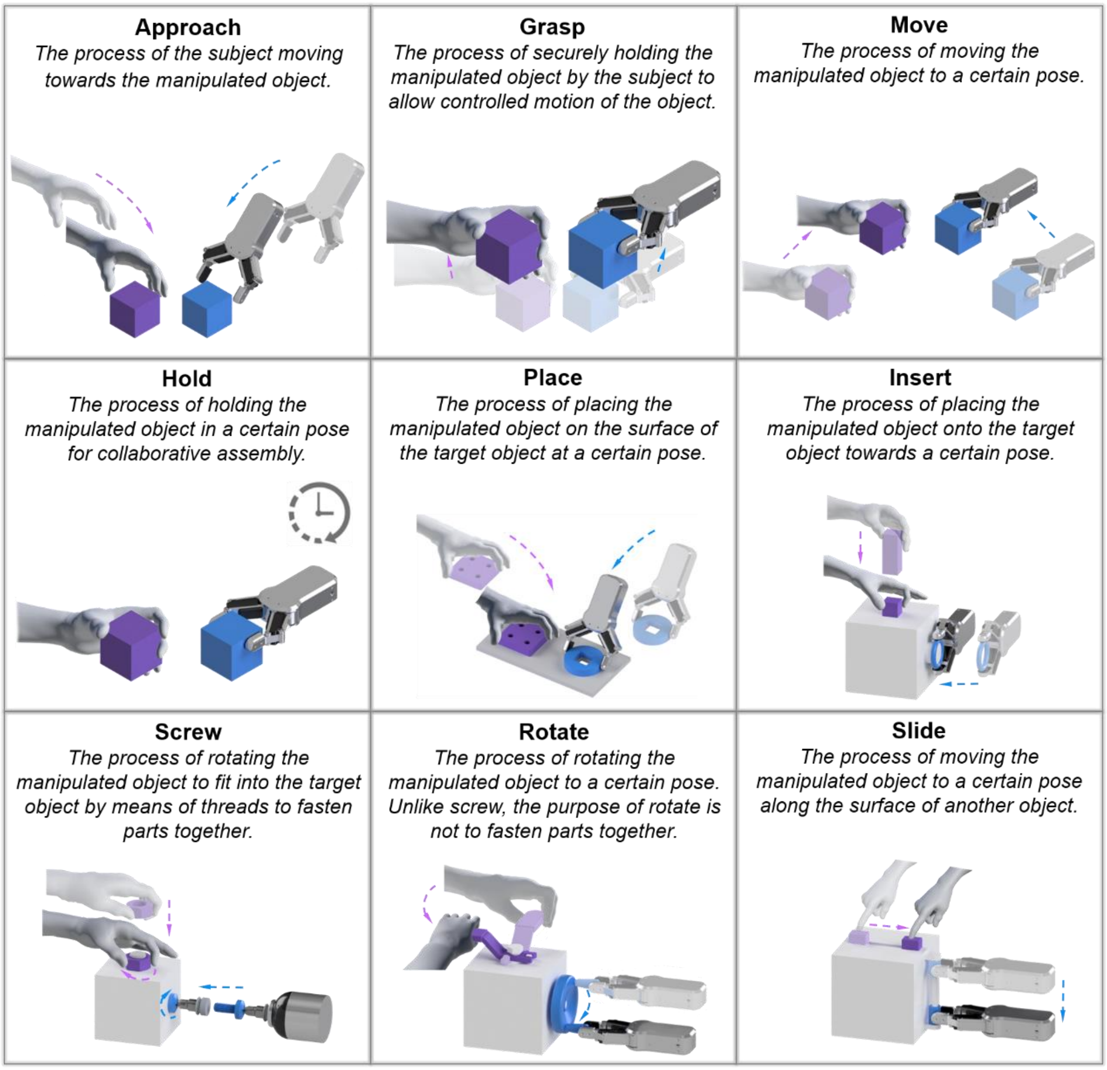

Fig. 1. Illustrations and definitions of nine atomic actions.

Table 1. Action parameters for the nine atomic actions

Action	Parameters	Reference
Approach	Potential approach poses of the subject: $[(T,R)^1,\dots,(T,R)^n]$ - The i-th approach direction: $T^i = (t_x^i, t_y^i, t_z^i)$ - The i-th approach posture: $R^i = (q_1^i, q_2^i, q_3^i, q_4^i)$	[1–3]

	- Reference system: MCS	
Grasp	Potential grasp poses of the subject: $[(T,R)^1,\ldots,(T,R)^n]$ - The i-th grasp direction: $T^i=(t_x^i,t_y^i,t_z^i)$ - The i-th grasp posture: $R^i=(q_1^i,q_2^i,q_3^i,q_4^i)$ - Reference system: MCS Grasp positions on MO: $(p_1,p_2,\ldots,p_n)$ - The i-th Grasp point: $p_i=(x_i,y_i,z_i)$ - Reference system: MCS Grasp force range: $[f_{min},f_{max}]$ - minimum force: f_{min} N - maximum force: f_{max} N	[4–11]
Move	Goal pose of MO: (T,R) - Direction: $T=(t_x,t_y,t_z)$ - Posture: $R=(q_1,q_2,q_3,q_4)$ - Reference system: WCS, TCS	[1,12]
Hold	Goal pose of MO: (T,R) - Direction: $T=(t_x,t_y,t_z)$ - Posture: $R=(q_1,q_2,q_3,q_4)$ - Reference system: WCS, TCS	[13]
Place	Goal pose of MO: (T,R) - Direction: $T=(t_x,t_y,t_z)$ - Posture: $R=(q_1,q_2,q_3,q_4)$ - Reference system: WCS, TCS	[14,15]
Insert	Goal pose of MO: (T,R) - Subassembly/part pose: - Direction: $T=(t_x,t_y,t_z)$ - Posture: $R=(q_1,q_2,q_3,q_4)$ - Reference system: TCS - Insert-feature (e.g., pin) poses: $[(T_1,R_1),\ldots,(T_n,R_n)]$ - Direction of the i-th insert-feature: $T_i=(t_x,t_y,t_z)$ - Posture of the i-th insert-feature: $R_i=(q_1,q_2,q_3,q_4)$ - Reference system: TCS Force: - Subassembly/part level force range: $[f_{min},f_{max}]$ N - Insert-feature level force range: $[f_1,\ldots,f_n]$ -The force range of i-th insert-area: $f^i=[f_{min}^i,f_{max}^i]$ N	[16–18]
Screw	Goal pose of MO: (T,R) - Direction: $T=(t_x,t_y,t_z)$ - Posture: $R=(q_1,q_2,q_3,q_4)$ - Reference system: TCS Torque range: $[m_{min},m_{max}]$ Nm Screw-in direction: clockwise or counterclockwise	[19,20]
Rotate	Contact points on MO: $[p_1,\ldots,p_n]$ - The i-th contact point: $p_i=(x_i,y_i,z_i)$ - Reference system: MCS Rotate angle: r - Reference system: TCS Goal pose of MO: (T,R) - Direction: $T=(t_x,t_y,t_z)$ - Posture: $R=(q_1,q_2,q_3,q_4)$ - Reference system: TCS	[21–23]

	Rotate direction: clockwise or counterclockwise	
Slide	Goal pose of MO: (T, R) - Direction: $T = (t_x, t_y, t_z)$ - Posture: $R = (q_1, q_2, q_3, q_4)$ - Reference system: TCS Sliding trajectory of MO: $(p_1, \dots, p_n)$ - The i-th trajectory point: $p_i = (x_i, y_i, z_i)$ - Reference system: TCS Contact points on TO: $[p_1, \dots, p_n]$ - The i-th contact point: $p_i = (x_i, y_i, z_i)$ - Reference system: MCS Force trajectory: $(f_1, \dots, f_n)$ - The i-th force: f_i N	[24,25]

*WCS: word coordinate system; MCS: manipulated object coordinate system; TCS: Target object coordinate system; MO: manipulated object; TO: target object.

Approach is the process of the subject moving towards the manipulated object. To describe this action requires the information of the subject's approaching direction and posture relative to the manipulated object [1–3], which can be described by the translation vector and rotation matrix of the goal 6D pose of the subject relative to the 6D pose of the manipulated object. The relative pose is under the manipulated object coordinate system. However, for a manipulated object, there may be multiple possible approach poses. Therefore we provide a list of potential approach poses: $[(T, R)^1, \dots, (T, R)^n]$, where $T^i = (t_x^i, t_y^i, t_z^i)$ is the i-th approach direction and $R^i = (q_1^i, q_2^i, q_3^i, q_4^i)$ is the i-th approach posture.

Grasp is the process of securely holding the manipulated object by the subject to allow controlled motion of the object [4]. Successful grasps can be force-closure [5] or form-closure [6], and the key parameters are the grasp pose, grasp position and the applied force. Grasp pose includes *grasp direction*, which is the direction of the subject approaching the manipulated object, and *grasp posture*, which is the rotation of the subject relative to the manipulated object. They are all independent of the subject and manipulated object properties, and are deterministic in the action description. The grasp direction and posture can be described by the translation vector and rotation matrix of the goal 6D pose of the subject relative to the 6D pose of the manipulated object, as (R, T). For a manipulated object, there may be multiple feasible grasp poses. Therefore, we provide the potential grasp poses as a list $[(T, R)^1, \dots, (T, R)^n]$. However, grasp position and applied force depend on the subject and manipulated object properties. There are various types of robot subjects, including hard or soft grippers with different fingers, suction disk and some special grippers, e.g., ball gripper and O-ring gripper [7–9]. These subjects may require different grasp positions for the same grasp task, thus the grasp positions are often untransferable between different types of robot subjects. However, the grasp position is a necessary part of a complete grasp action description, which is indispensable in some specific cases. Therefore, the taxonomy should include the grasp position information. The grasp position can be described by the grasp point, which can be *contact point* [4] or *grasp center point* [10]. To meet the needs of different applications, this taxonomy only provides the data interface of grasp points, and users can flexibly define grasp points according to their own needs. The grasp point is the point on the manipulated object, and it can be described by the three-dimensional point under the manipulated object coordinate system, as $(p_1, p_2, \ldots, p_n)$, where $p_i = (x_i, y_i, z_i)$ is the i-th grasp point. Same with grasp point, we only provide the data interface of grasp force. The grasp force refers to the force that each finger applies on the object to grasp it securely without slipping or causing damage, which corresponds to the contact points [11]. The force has direction and magnitude. Since the direction is the normal vector of the contact point, it has already been constrained. We only provide the magnitude of the force. The force requirement is often a range to ensure that the object does not slip and is not damaged. Therefore, the action parameter of grasp force is a numerical range $[f_{min}, f_{max}]$ N.

Move is the process of moving the manipulated object to a certain pose. Therefore, the key descriptive information is the goal 6D pose of the manipulated object [1,12,26]. However, the pose can be absolute or relative, which depends on the specific application. If the action after Move is Hold, the object's pose is not constrained by another object, so the pose is absolute in the world coordinate. However, if the action after the Move is Insert, Screw or Place, the pose should be constrained to the relative pose of the manipulated object to the target object.

Hold is the process of holding the manipulated object in a certain pose for collaborative assembly [13]. During this period of time, holding the manipulated object may be only for waiting the completion of other assembly actions, or the held manipulated object can be taken as the target object, and other subjects can assemble other manipulated objects with it. No matter what the purpose of the action is, the parameter is the 6D pose of the object, and the pose is absolute in the world coordinate.

Place is the process of placing the manipulated object on the surface of the target object at a certain pose. Therefore, the key descriptive information is the 6D pose of the manipulated object relative to the target object [14,15,26]. The difference between

Place and Insert is that Place does not involve the mating of two objects. Therefore, there is no force constraint.

Insert is the process of placing the manipulated object onto the target object towards a certain pose. Describing the action of inserting the manipulated object onto the target object requires the information of the relative pose between the two objects after the insertion, as well as the force range required during the insertion [16–18,26]. However, there may be multiple insert-features (e.g., pins) corresponding to multiple inserted-features (e.g., holes) on one subassembly/part. Therefore, when describing the relative pose, there are two levels: one is subassembly/part level, which describes the 6D pose of manipulated subassembly/part relative to target subassembly/part as (T, R), and the other is assemblable feature level, which describes the 6D pose of each manipulated insert-feature relative to the corresponding target inserted-feature as $[(T_1, R_1), \dots, (T_n, R_n)]$. Similarly, the description of the applied force range should also have two levels: the subassembly/part level $[f_{min}, f_{max}]$ N and the assemblable feature level $[f_1, \dots, f_n]$, where $f^i = [f^i_{min}, f^i_{max}]$ N. The direction of the force is along the axial of the hole, so it is only necessary to describe the magnitude of the force. The reason why the requirements of pose and force are divided into two levels is that applying the insert force at one time and at one point may not ensure that all insert-features are inserted successfully. Therefore, the hierarchical representation can facilitate the inspection of the insertion results, and it is also convenient for users to manage the force distribution according to their own robot capability.

Screw is the process of rotating the manipulated object to fit into the target object by means of threads to fasten parts together. Threaded object can be categorized into inner-threaded object and outer-threaded object, but both of them can be manipulated object or target object. There may be multiple screw-in/screwed-in features on one subassembly/part. To completely describe a Screw action, it requires the information of the goal 6D pose of the manipulated object relative to target object [19], as (T, R). The purpose of Screw is to fasten two/more objects together. To ensure the fastening quality, it is usually necessary to specify the preload [20]. The preload acting on the manipulated object and the target object is a pair of forces with the same magnitude and opposite direction. Since the direction is along the central axis of the thread, it has been constrained already. Only the magnitude of the preload needs to be constrained, so the preload requirement here is a constant. Because controlling torque is easier than controlling force when screwing, the preload requirement can be constrained by the tightening torque acting on the manipulated object, and it is also a constant. Because the torque requirement usually is a torque range, we provide the torque range as $[m_{min}, m_{max}]$ Nm. In addition, another important descriptive information is the screw-in direction. Threads can be lefthanded or right-handed, thus the screw-in direction can be clockwise or counterclockwise. The screw-in direction is also the direction of applying the torque.

Rotate is the process of rotating the manipulated object to a certain pose. Unlike screw, the purpose of rotate is not to fasten parts together. The key descriptive information is the goal pose of the manipulated object relative to the target object [19,21,23], as (T, R). It is sometimes necessary to describe the contact points between the subject and the manipulated object. The contact points are the three-dimensional points in the manipulated object coordinate system, as $(p_1, p_2, \dots, p_n)$, where $p_i = (x_i, y_i, z_i)$ is the i-th contact point. Another important descriptive information is the rotate angle – the angle of rotating the manipulated object. The rotate angle is a constant r.

Slide is the process of moving the manipulated object to a certain pose along the surface of the target object. The key descriptive information is the goal pose of the manipulated object, sliding trajectory, contact points and sliding force [24,25]. The goal pose is the 6D pose of the manipulated object relative to the target object, as (T, R). The sliding trajectory is a line on the surface of the target object, which can be represented by a series of three-dimensional points in the target object coordinate system, as $(p_1, p_2, \dots, p_n)$, where $p_i = (x_i, y_i, z_i)$ is the i-th trajectory point. On this trajectory, the initial position of the object can be anywhere, but the goal position is fixed. The contact points between the subject and the manipulated object can be represented by the three-dimensional points in the manipulated object coordinate system, as $(p_1, p_2, \dots, p_n)$, where $p_i = (x_i, y_i, z_i)$ is the i-th contact point. The direction of the sliding force applied on the manipulated object should be along the tangent direction of the sliding trajectory. Since the magnitude of the force required at different positions on the sliding trajectory may be different, the data interface we provide is a force trajectory, which means each trajectory point corresponds to a force magnitude. We provide the force trajectory as $(f_1, \dots, f_n)$ where f_i N is the force magnitude of the i-th force trajectory point.

Reference:

[1] J. Jeon, H. Jung, F. Yumbla, T.A. Luong, H. Moon, Primitive Action Based Combined Task and Motion Planning for the Service Robot, Front. Robot. AI. 9 (2022). https://doi.org/10.3389/frobt.2022.713470.

[2] X.F. Zha, Optimal pose trajectory planning for robot manipulators, Mech.

Mach. Theory. 37 (2002) 1063–1086. https://doi.org/10.1016/S0094-114X(02)00053-8.

[3] A.T. Le, M. Guo, N. van Duijkeren, L. Rozo, R. Krug, A.G. Kupcsik, M. Burger, Learning Forceful Manipulation Skills from Multi-modal Human Demonstrations, in: 2021 IEEE/RSJ Int. Conf. Intell. Robot. Syst., IEEE, 2021: pp. 7770–7777. https://doi.org/10.1109/IROS51168.2021.9636828.

[4] S. Caldera, A. Rassau, D. Chai, Review of deep learning methods in robotic grasp detection, Multimodal Technol. Interact. 2 (2018). https://doi.org/10.3390/mti2030057.

[5] V.-D. Nguyen, Constructing Force-Closure Grasps, Int. J. Rob. Res. 7 (1988) 3–16. https://doi.org/10.1177/027836498800700301.

[6] A. Bicchi, On the Closure Properties of Robotic Grasping, Int. J. Rob. Res. 14 (1995) 319–334. https://doi.org/10.1177/027836499501400402.

[7] G. Du, K. Wang, S. Lian, K. Zhao, Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review, Springer Netherlands, 2021. https://doi.org/10.1007/s10462-020-09888-5.

[8] Z. Wang, Y. Torigoe, S. Hirai, A Prestressed Soft Gripper: Design, Modeling, Fabrication, and Tests for Food Handling, IEEE Robot. Autom. Lett. 2 (2017) 1909–1916. https://doi.org/10.1109/LRA.2017.2714141.

[9] Z. Samadikhoshkho, K. Zareinia, F. Janabi-Sharifi, A Brief Review on Robotic Grippers Classifications, 2019 IEEE Can. Conf. Electr. Comput. Eng. CCECE 2019. (2019) 1–4. https://doi.org/10.1109/CCECE.2019.8861780.

[10] J. Redmon, A. Angelova, Real-time grasp detection using convolutional neural networks, Proc. - IEEE Int. Conf. Robot. Autom. 2015-June (2015) 1316–1322. https://doi.org/10.1109/ICRA.2015.7139361.

[11] A. Bicchi, V. Kumar, Robotic grasping and contact: a review, in: Proc. 2000 ICRA. Millenn. Conf. IEEE Int. Conf. Robot. Autom. Symp. Proc. (Cat. No.00CH37065), IEEE, n.d.: pp. 348–353. https://doi.org/10.1109/ROBOT.2000.844081.

[12] M. Raessa, D. Sánchez, W. Wan, D. Petit, K. Harada, Teaching a robot to use electric tools with regrasp planning, CAAI Trans. Intell. Technol. 4 (2019) 54–63. https://doi.org/10.1049/trit.2018.1062.

[13] W. He, J. Li, Z. Yan, F. Chen, Bidirectional Human–Robot Bimanual Handover of Big Planar Object With Vertical Posture, IEEE Trans. Autom. Sci. Eng. 19 (2022) 1180–1191. https://doi.org/10.1109/TASE.2020.3043480.

[14] S.D. Han, S.W. Feng, J. Yu, Toward Fast and Optimal Robotic Pick-and-Place on a Moving Conveyor, IEEE Robot. Autom. Lett. 5 (2020) 446–453. https://doi.org/10.1109/LRA.2019.2961605.

[15] B. Tipary, G. Erdős, Generic development methodology for flexible robotic

pick-and-place workcells based on Digital Twin, Robot. Comput. Integr. Manuf. 71 (2021) 102140. https://doi.org/10.1016/j.rcim.2021.102140.

[16] B. Niu*, C. Wang*, C. Liu, Tolerance-Guided Policy Learning for Adaptable and Transferrable Delicate Industrial Insertion, (2021). http://arxiv.org/abs/2108.02303.

[17] T. Tang, H.C. Lin, Y. Zhao, Y. Fan, W. Chen, M. Tomizuka, Teach industrial robots peg-hole-insertion by human demonstration, IEEE/ASME Int. Conf. Adv. Intell. Mechatronics, AIM. 2016-Septe (2016) 488–494. https://doi.org/10.1109/AIM.2016.7576815.

[18] A. Kramberger, A. Gams, B. Nemec, D. Chrysostomou, O. Madsen, A. Ude, Generalization of orientation trajectories and force-torque profiles for robotic assembly, Rob. Auton. Syst. 98 (2017) 333–346. https://doi.org/10.1016/j.robot.2017.09.019.

[19] R.L.A. Shauri, K. Saiki, S. Toritani, K. Nonami, Sensor Integration and Fusion for Autonomous Screwing Task by Dual-Manipulator Hand Robot, Procedia Eng. 41 (2012) 1412–1420. https://doi.org/10.1016/j.proeng.2012.07.329.

[20] E. Berger, S. Grehl, D. Vogt, B. Jung, H. Ben Amor, Experience-based torque estimation for an industrial robot, in: 2016 IEEE Int. Conf. Robot. Autom., IEEE, 2016: pp. 144–149. https://doi.org/10.1109/ICRA.2016.7487127.

[21] S.R. Ahmadzadeh, P. Kormushev, R.S. Jamisola, D.G. Caldwell, Learning reactive robot behavior for autonomous valve turning, in: 2014 IEEE-RAS Int. Conf. Humanoid Robot., IEEE, 2014: pp. 366–373. https://doi.org/10.1109/HUMANOIDS.2014.7041386.

[22] A.S. Morgan, D. Nandha, G. Chalvatzaki, C. D’Eramo, A.M. Dollar, J. Peters, Model Predictive Actor-Critic: Accelerating Robot Skill Acquisition with Deep Reinforcement Learning, Proc. - IEEE Int. Conf. Robot. Autom. 2021-May (2021) 6672–6678. https://doi.org/10.1109/ICRA48506.2021.9561298.

[23] T. Narita, O. Kroemer, Policy Blending and Recombination for Multimodal Contact-Rich Tasks, IEEE Robot. Autom. Lett. 6 (2021) 2721–2728. https://doi.org/10.1109/LRA.2021.3061982.

[24] T. Eiband, D. Lee, Identification of Common Force-based Robot Skills from the Human and Robot Perspective, in: 2020 IEEE-RAS 20th Int. Conf. Humanoid Robot., IEEE, 2021: pp. 507–513. https://doi.org/10.1109/HUMANOIDS47582.2021.9555681.

[25] V.K. Origanti, T. Eiband, D. Lee, Automatic Parameterization of Motion and Force Controlled Robot Skills, in: 2022: pp. 66–78. https://doi.org/10.1007/978-3-030-97672-9_7.

[26] J.O. Huckaby, H.I. Christensen, A taxonomic framework for task modeling and knowledge transfer in manufacturing robotics, in: Work. Twenty-Sixth AAAI Conf. Artif. Intell., 2012.