2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Reducing the Teleoperator's Cognitive Burden for Complex Contact Tasks Using Affordance Primitives

Adam Pettinger, Cassidy Elliott, Pete Fan, Mitch Pryor†
Walker Department of Mechanical Engineering
The University of Texas at Austin
Austin, Texas 78712

Abstract— **Using robotic manipulators to remotely perform real-world complex contact tasks is challenging whether tasks are known (due to uncertainty) or unknown *a priori* (lack of motion waypoints, force profiles, etc.). For known tasks we can integrate and utilize Affordance Templates with a selective compliance jogger to remotely perform high dimensional velocity/force tasks - such as turning valves, opening doors, etc. Affordance Templates (ATs) contain virtual visual representations of task-relevant objects and waypoints for interacting with visualized objects. Operators and/or developers align pre-defined ATs with real-world objects to complete complex tasks, potentially reducing the operator's input dimension to a single initiation command. In this work, we integrate a compliant controller with existing ATs to reduce the operator's burden by 1) reducing the dimension of commanded inputs, 2) internally managing contact forces even for complex tasks, and 3) providing situational awareness in the task frame. Since not all tasks can be modeled for general teleoperation, we also introduce *Affordance Primitives* which reduce the command dimensionality of complex spatial tasks to as low as 1-dimensional input gestures as demonstrated for this effort. To enable reduction of the command input's dimension, the same compliant jogger used to robustly handle uncertainty with ATs is used with Affordance Primitives to autonomously maintain force constraints associated with complex contact tasks. Both Affordance Templates and Affordance Primitives - when used in tandem with a compliant jogger - provide a safe, intuitive, and efficient teleoperation system for general use including using primitives to easily develop new Affordance Templates from newly completed teleoperation tasks.**

I. INTRODUCTION

Robots have been used to perform tasks in hazardous or remote environments since the 1940s [1], [2]. Robotics literature in the area of improving human-in-the-loop task performance is abundant including [3–6]. Recent development of Affordance Templates (ATs) by NASA and TRACLabs provide a "task representation and execution framework" [7] designed to help execute manipulation tasks with operator help. Affordance templates define a set of End-Effector (EEF) waypoints relative to a 3D model of relevant task objects. ATs were featured in the DARPA Robotics Challenge as a supervised autonomy control scheme to perform tasks such as turning valves and manipulating power tools. The templates were aligned with task objects by an operator utilizing an interactive marker in RViz [8], but their current use has limitations. The waypoints generated by an AT are left to the user to satisfy. Most affordance[1] implementations use *MoveIt!* [10] for planning-execution of the trajectories. However, *MoveIt!* is not inherently good at performing the type of contact tasks ATs enable: planning around collision objects that the operator only *sometimes* wants to touch is tricky and requires back-end work for the user, basic trajectory planning is guaranteed to pass through all waypoints but may not maintain a smooth or continuous trajectory, and *MoveIt!* execution relies on position requirements that may be violated due to interaction with the environment. Additionally, since ATs rely on sensor data for the operator to align the template with the real-world object, error in the data can make alignment difficult and introduces positional uncertainty into the motion planner. ATs have been implemented for humanoid robots such as Valkyrie and Atlas [11] and overcome these problems with careful alignment/planning from the user as well as mechanically compliant hardware.

Implementing ATs for robots without built-in compliance is more challenging since modelling errors in the template, uncertainty in the sensor data, and errors in template alignment must all be addressed by the controller. Here we present a robust control scheme for complex[2] contact tasks that can be used both during teleoperation and with Affordance Templates to complete tasks in four operational scenarios.

1) Pure teleoperational control using the *MoveIt!* planning-execution framework, EEF jogging[3] [12], or others
2) Compliant, motion-based teleoperated jogging that is robust to uncertainties in both the environment and with respect to user input
3) Compliant AT execution that is robust to uncertainty in the environment including template alignment errors
4) Pure AT execution using a motion planner such as *MoveIt!* to plan through a sequence of waypoints

The current state-of-the-art for AT use is scheme 4, and the *de facto* standard for ROS control is scheme 1 with *MoveIt!*.

†{adam.pettinger,cmelliott2,petefan.7, mpryor}@utexas.edu

---

[1] The term "affordance" is derived from the term's use in biology to refer to what an environment provides or furnishes an animal. It summarizes how a particular animal can interact with a particular environment [9].

[2] Complex implies the contact forces are neither constant or limited to a single direction.

[3] Here "jogging" is considered to be streaming EEF velocity commands to the controller which calculates joint velocities, using an Inverse Jacobian method in this work.

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

Our efforts address schemes 2 and 3, which enable non-expert users to more easily control the robot. This paper presents work to better operate in scheme 2, and capture the necessary motion parameters shared between schemes 2 and 3 as Affordance Primitives (APs), which then enable affordance template generation directly from task execution.

The motivating project for this work uses a dual-armed mobile manipulator shown in Fig. 1 to perform basic maintenance, inspection, and emergency tasks in a liquid natural gas (LNG) facility. We evaluate the developed capabilities by completing a *double-block-&-bleed* task that involves closing 3 rotary valves in a specified order. We use a compliant jogger with various motion-controller input devices to turn the valves quicker and with fewer faults. The remainder of this paper focuses on the specific task described, but the approach is applicable to a variety of other contact tasks, manipulators, and user interface schemes. Examples of other tasks include opening doors, aligning a manipulator's tool changer, plugging in electrical sockets and flipping switches and pushing buttons designed for humans. After a review of previous works with Affordance Templates and compliant control, we discuss Affordance Primitive development and controller implementation. We then present our test setup and experimental results before our conclusions and future work.

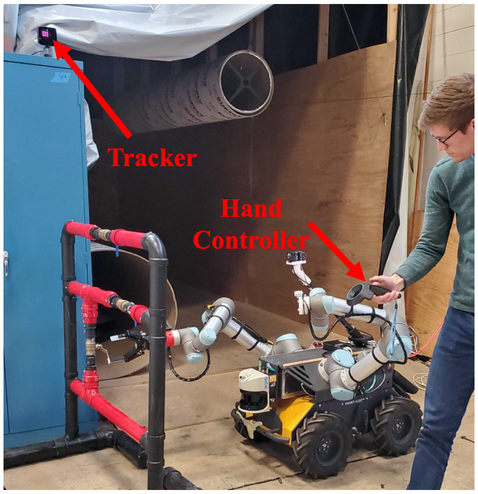

Fig. 1. Components utilized to evaluate complex task completion using Affordance Primitives including a mobile manipulator with F/T sensor and gripper, user input devices (motion capture controller, 3D mouse), and *double-block-&-bleed* prop

II. BACKGROUND AND PRIOR WORK

A. *Affordance Templates*

The two main Affordance Template software packages, Craftsman and *UseIt!*, were developed by TRACLabs and NASA respectively to help automate task execution. Predecessors to ATs included MIT's Object Template Description Format (OTDF) [13] and IHMC's Coactive Design method [14]. ATs built upon these methods enable a variety of adjustments to the affordance object, increasing operator interaction [7]. While NASA's *UseIt!* is proprietary, a version of TRACLabs' Craftsman is open-source with several papers detailing its AT structure [7], [15–18].

Each AT consists of a display object with an associated sequence of EEF waypoints specified in the object coordinate frame, shown in Fig. 2. Pure ATs do not allow for shared control during the task execution; once the user makes their initial adjustments to the AT waypoints and executes the plan, they have no further control over the motion. Thus, if the execution fails, the object must be reset and the task re-attempted. The combination of a compliance jogger with ATs increases the likelihood of completing the task.

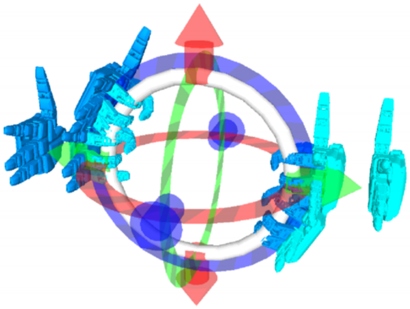

Fig. 2. An example of an AT for grasping and turning a wheel [7].

Hart et al. identified force-based tasks as a vital area of future development for ATs to be able to complete "sophisticated manipulation tasks in real-world contexts" [7]. As of their current design, ATs are instantiated only with spatial data, and do not contain the necessary information to derive a task's associated motion and compliance parameters. The *Affordance Primitives* developed and reviewed in Section III-C address this issue.

Other ongoing work with ATs includes the use of affordance wayfields to quantify how well certain affordances can be performed on an object [19] and the automatic registration of affordances to manipulation objects [20]. Affordance wayfields remove the dependency on waypoints to instead focus on EEF motions. Wayfields compensate for AT limitations including excessive complexity for intricate tasks, lack of robot pose control, and inability to control movement between waypoints, account for reachability, or handle obstacles between waypoints [19]. Automatically registering ATs to the real-world objects would increase task speed and reduce position error due to misaligned templates. Recent research efforts show the desire to use ATs to simplify the user interface, but also the need to improve execution for increasingly complicated tasks.

B. *Compliant Control*

Most AT demonstrations utilized compliant hardware, which limits their use for stiff manipulators that are often simpler, cheaper, and more precise. These systems must address compliance issues in the controller. Compliant (e.g. force, impedance) control of manipulators has been an active area of research since the 1980's [21–23], especially in the context of performing contact tasks, allowing them to better manage contact forces with the environment and improving safety and performance during contact. In [24], Walker notes that with error in the manipulator position, "the contact or grasp may become essentially an impact or collision" and that "it is desired that the manipulation be as gentle as possible" to avoid damaging the robot or environment.

Hybrid control [25], [26] was also introduced in the 1980's, as the concept of controlling unique dimensions of the manipulator's output space with different control laws, mostly force or position control. Impedance control [27] places a "virtual impedance" between the manipulator's actual position and desired position. Then the forces acting on the manipulator are used to control the robot's desired position based on the impedance. In this work, we use an impedance controller [12] with a hybrid-like ability to toggle compliance in each output DoF.

III. IMPLEMENTATION

A. Compliant Cartesian Jogging

The teleoperation controller utilizes the well-known inverse Jacobian method [28]. The jogger presented in [12] is currently available as a ROS package for teleoperated velocity control[4]. The packaged jogger slows down as the manipulator approaches a kinematic singularity and prevents self collisions. It allows for redundant manipulators including either $n > 6$ joints or reduced task space $m < \mathbb{R}^6$. The input to the jogger is a Cartesian EEF velocity that can be received from multiple interfaces including joysticks, motion trackers, camera-captured gestures, voice commands, etc. The jogger outputs joint velocities and (optionally) positions. By default, we use the Singular Value Decomposition (SVD) pseudoinverse [29] which allows for redundant manipulators and is more stable near singularities than the Moore-Penrose pseudoinverse. Formally:

$$\dot{\mathbf{q}} = J^{\dagger}\dot{\mathbf{x}} \tag{1a}$$

$$J^{\dagger} = V\Sigma^{I}U^{T} \tag{1b}$$

$$U\Sigma V^{T} = J \tag{1c}$$

where $\dot{\mathbf{q}} \in \mathbb{R}^{n\times 1}$ are the joint velocities for an n jointed manipulator. $\dot{\mathbf{x}} \in \mathbb{R}^{m\times 1}$ is the EEF velocity in an m sized output space, $J \in \mathbb{R}^{m\times n}$ is the Jacobian, and $J^{\dagger}$ its pseudoinverse found using the SVD shown in (1c) where U and V are the unitary matrices, Σ is a diagonal matrix of singular values, and Σ^{I} its element-wise inverse.

During runtime, the jogger calculates the joint velocities $\dot{\mathbf{q}}_{\mathbf{jog}}$ for a desired EEF command using (1a). Simultaneously, the compliant controller calculates the "impedance" joint velocities $\dot{\mathbf{q}}_{\mathbf{comp}}$ due EEF contact forces. The compliant EEF velocity $\dot{\mathbf{x}}_{\mathbf{comp}}$ ($\dot{\mathbf{x}}$ in (1a)) is found using an impedance control law

$$\dot{\mathbf{x}}_{\mathbf{comp}} = \mathbf{K}^{-1}(\xi - \xi_{apply}) - \mathbf{B}^{-1}\dot{\xi} \tag{2}$$

where $\xi \in \mathbb{R}^{m\times 1}$ is the external wrench applied to the EEF, $\xi_{apply} \in \mathbb{R}^{m\times 1}$ is the wrench we desire the manipulator apply to the environment, and $\mathbf{K}$ and $\mathbf{B} \in \mathbb{R}^{m\times m}$ are diagonal matrices of the impedance stiffness and damping relating the external forces/torques to EEF velocity. We restrict $\mathbf{K}$ and $\mathbf{B}$ to be diagonal to reduce the number of parameters. The jogging command and compliance feedback are combined as

$$\dot{\mathbf{q}}_{\mathbf{total}} = J^{\dagger}diag(\nu)\dot{\mathbf{x}}_{\mathbf{jog}} + J^{\dagger}diag(\mu)\dot{\mathbf{x}}_{\mathbf{comp}} \tag{3}$$

where $\dot{\mathbf{x}}_{\mathbf{jog}}$ is the user input and $\dot{\mathbf{x}}_{\mathbf{comp}}$ is the compliance calculated with (2). The vectors ν and μ are m-sized Boolean vectors where $\nu_i, \mu_i \in \{0, 1\}$. We use ν and μ to selectively "turn off" jogging control or compliance along dimensions of the manipulator output space, enabling non-precise control input in unimportant directions to be ignored. For the duration of this paper, we discuss the user's "input space" as aligned with the task space, but reduced in dimensionality by ν.

B. Motion-Controller Jogging

Prior work with the jogger used a SpaceMousePro 6-DOF input device [12]. The SpaceMouse worked well to demonstrate the jogger's effectiveness, but was not intuitive for new users. Room-scale virtual reality systems such as the HTC Vive offer a new input method for teleoperated jogging that track the pose of one or more handheld controllers using an external infrared camera system and gyroscopes located in each controller (Fig. 1). One of the greatest advantages of using room-scale motion controllers for manipulator jogging is that EEF displacement is proportional to controller displacement. An operator performing complex tasks using the motion controllers can simply trace the desired EEF path, whereas a joystick operator must command the velocity at every point required to follow the path. The difference in input methods becomes more pronounced with complicated paths that involve tracing curves and rotating the EEF concurrently.

To obtain $\dot{\mathbf{x}}$ in (1a) from the motion controllers, we track the change in controller pose as

$$\dot{\mathbf{x}}_{\mathbf{trans}} = k_{trans}\frac{\Delta C_{pos}}{t_{poll}} \tag{4a}$$

$$\dot{\mathbf{x}}_{\mathbf{rot}} = k_{rot}\frac{\Delta C_{rot}}{t_{poll}} \tag{4b}$$

where $\dot{\mathbf{x}} = [\dot{\mathbf{x}}_{\mathbf{trans}}, \dot{\mathbf{x}}_{\mathbf{rot}}]^{T}$. t_{poll} is the sampling period and ΔC_{pos} and ΔC_{rot} are the change in controller position and rotation. Scaling constants k_{trans} and k_{rot} can dynamically adjust the jog velocity, providing the advantages discussed in [30]. It is possible, but not necessary, to find k_{trans} and k_{rot} such that the EEF follows the motions of the operator $1:1$.

Motion controllers enable one-handed operation allowing a user to control multiple manipulators, and have a large set of buttons that can be mapped to jogger settings such as modifying the jogging frame, adjusting k_{trans} on the fly, or

---

[4] See `ros-planning.github.io/moveit_tutorials/doc/arm_jogging/arm_jogging_tutorial.html`

actuating a gripper. The disadvantages of motion controllers include additional setup time and space, input signal noise.

C. Affordance Primitives

If a template does not exist to define the task affordances, then primitives are used to identify operational parameters for the compliant jogger and intuitively reduce the input space ($\mathbb{R}_i$) relative to the task space ($\mathbb{R}_m \leq 6$). Each primitive is task dependent and tuning each to a new task is non-trivial. To easily perform complex contact tasks either with teleoperation or Affordance Templates, it is necessary to quantify these parameters for a wide variety of tasks. In this work we do not present a solution to intelligently pick and test parameters, and instead focus on identifying and formalizing which parameters to use to enable completing tasks with teleoperation and thus identify parameters for future ATs for similar tasks. We call this parameter set *Affordance Primitives*, which define the allowable (or afforded) behaviors of a manipulator performing a task while in contact with the environment.

For a manipulator with n joints operating in m DoF output space, we define an Affordance Primitive as a tuple where all units are SI.

- $k \in \mathbb{R}^m$: impedance "stiffness" relating applied wrench to EEF velocity and forming the diagonal matrix $\mathbf{K}$ from (2)
- $\beta \in \mathbb{R}^m$: impedance "damping" relating the time derivative of applied wrench to EEF velocity and forming the diagonal matrix $\mathbf{B}$ from (2)
- $\xi_{apply} \in \mathbb{R}^m$: desired applied wrench in (2), given in N and $N \cdot m$
- $\xi_{max} \in \mathbb{R}^m$: maximum allowable force/torque in each direction in N and $N \cdot m$
- $F_{max} \in \mathbb{R}^1$: maximum combined force in N
- $\tau_{max} \in \mathbb{R}^1$: maximum combined torque in $N \cdot m$
- $\Delta x_{max} \in \mathbb{R}^m$: maximum allowable displacement from compliance
- $\dot{x}_{max} \in \mathbb{R}^m$: maximum allowable EEF velocity
- $\dot{q}_{max} \in \mathbb{R}^n$: maximum allowable joint velocities in (3)
- $\nu \in \mathbb{Z}_2^m$: jogging control dimensions allowed in (3)
- $\mu \in \mathbb{Z}_2^m$: compliant dimensions in (3)

The result is a $8m + n + 2$ tuple defining the compliance controller's interaction with an externally applied wrench. Note that ν and μ reduce the input and compliance space respectively, and serve to conceptually aid the user and may be ignored in any generated ATs. Affordance Primitive parameter values are most easily intuited from previously defined ATs for the task space (i.e. a particular robot in a particular environment interacting with similar objects), but can also be defined by the operator, or be pulled from a set of robot-specific default values. For the testing discussed in Section IV, the AP parameters were chosen experimentally since such data did not exist until the task was completed. Automating the parameter selection would be an area for future work.

IV. EXPERIMENTATION

The elements above are combined into the control scheme shown in Fig. 3. The controller is implemented as a series of independent ROS nodes running across multiple computers.

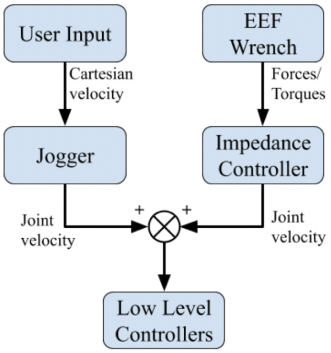

Fig. 3. Compliant jogger control overview

The "User Input" is from the motion capture device discussed in Section III-B. The jogger reviewed in Section III-A determines the joint velocities which are provided by the Impedance Controller using F/T sensor data. The summation junction represents a *ros_control*[5] controller that enforces position/velocity limits and other safety considerations. The AP parameters dictating controller behavior for specific tasks were discussed in Section III-C.

The task selected for user evaluation is a *double-block-&-bleed*, where rotary ball valves are turned in a specific sequence to *block* out a segment of pipe which is then *bled* for inspection or repair. We evaluate the proposed method by giving novice users a system overview and asking them to remotely turn valves. The valve, gripper, and motion controller are shown in Fig. 4 and 5.

First, we describe the task including the Affordance Primitives used to reduce the operator's burden. Each valve manipulation is split into sub-tasks: 1) moving EEF into position and grasping the valve, and 2) turning the valve 90° to close it. The AP parameters for the sub-tasks are shown in Table I. For tasks where the same robot is interacting with the same device, this is expected and simplifies their development. Other AP parameters include a maximum force F_{max} of 88 N, and a maximum torque τ_{max} of 50 Nm. The velocity limits $\dot{q}_{max}$ and $\dot{x}_{max}$ were set to the manipulator's limits, but for other tasks it may be desirable to use more conservative values.

The grasp sub-task (Fig. 4) included additional parameters shown in Table II, where the user's input space dimensionality matches the full task space ($\mathbb{R}_m^6$) so they can align the gripper with the valve. The compliance dimensions were chosen to align the gripper as it closes, makes contact, and grasps the handle. In this case, the compliance ensures the grasp occurs no matter how the operator orients the valve handle inside the gripper prior to closing. As long as the valve handle is between the gripper's fingers, compliance

---

[5] See `https://wiki.ros.org/ros_control`

TABLE I

AP PARAMETERS SHARED WITH GRASP AND TURN

Parameter	X	Y	Z	Roll	Pitch	Yaw
k	8000	1000	1000	5	40	60
β	50000	10000	10000	300	600	600
ξ_{apply}	0	0	0	0	0	0
ξ_{max}	80	80	80	60	60	60
Δx_{max}	0.15	0.15	0.05	$\pi/2$	$\pi/16$	$\pi/16$

will close the gripper while accommodating any offset or orientation errors without creating undue force on the valve handle. Thus for the actual grasping, the user input is a single button press ($\mathbb{R}^6 \rightarrow \{0, 1\}$), and they are relieved of the burden of managing the grasp to avoid large forces between the robot and handle.

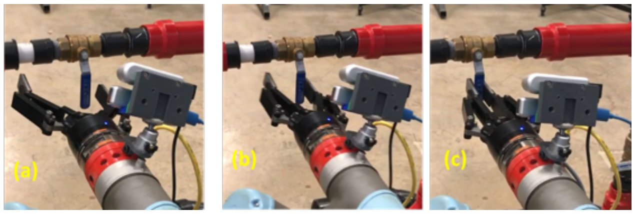

Fig. 4. In (a), the user has located gripper around the valve with significant offset and rotational errors. In (b), the gripper has started to close and makes contact with the valve with one finger before (c) the affordance compliance parameters assure the controller continues to grasp the handle while compliantly correcting the operator's positioning errors

TABLE II

AP PARAMETERS FOR VALVE GRASP

Parameter	X	Y	Z	Roll	Pitch	Yaw
ν	1	1	1	1	1	1
μ	1	1	0	1	0	0

Once grasped, the second sub-task, closing the valve (Fig. 5), is attempted. The ν and μ parameters are changed as shown in Table III. Here, the user's input space was reduced to Y-axis translation and Roll about the X-axis ($\mathbb{R}_i^6 \rightarrow \mathbb{R}_i^2$). With compliance also enabled in the Roll direction, the manipulator is able to move tangentially to the valve handle's arc in order to turn the valve while the Roll compliance keeps the EEF's Y-axis translation aligned with the tangent. Relying on the compliance removes the need to know the valve's turn radius, and the X-axis translation compliance overcomes misalignment between the valve's axis of rotation and the EEF Roll axis.

TABLE III

AP PARAMETERS FOR VALVE TURN

Parameter	X	Y	Z	Roll	Pitch	Yaw
ν	0	1	0	1	0	0
μ	1	0	0	1	0	0

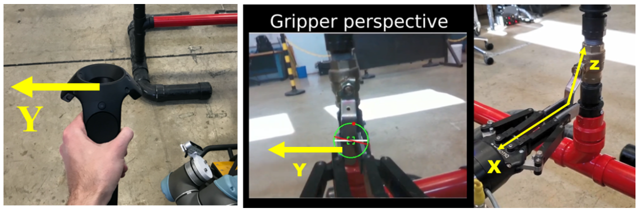

Fig. 5. Left, the user simply moves (a swipe gesture) the controller to the left, while (middle, right) the developed controller utilizes compliance to assure the gripper correctly tracks the rotation and elevation in the grasp point as the valve is closed.

Users were asked to perform the task three times: first without compliance or reduced input space, then with compliance parameters and directions μ from the tables but no reduction in input space, and finally with full compliance and the reduced input space described above. For each test, we evaluate the task with the following criteria:

- Completion: Was the user able to fully perform the task?
- Time: How long did it take the user to complete or abandon the task?
- Number of safety faults: The UR5 controller automatically e-stops under high loads. Tracking the safety faults let us track how often high loads were applied in a test.
- User opinion of difficulty: How difficult did the user rate the task for each input method?

During each run, the user was allowed to use the SpaceMouse, motion controllers, or both. In fact, some users preferred one while others used both to good effect. The results across the three runs are given in Fig. 6 and Table IV.

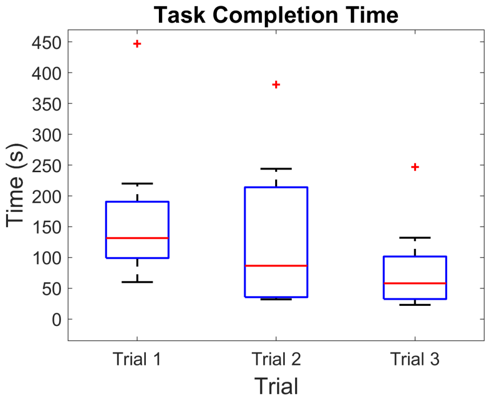

Fig. 6. Users more quickly and reliably completed the *double-block-&-bleed* task when utilizing Affordance Primitives to reduce their cognitive burden (n=8).

TABLE IV

TESTING RESULT AVERAGES

Trial	Time (s)	Safety Faults	Difficulty (1-5)
Trial 1	168.6	4.9	4
Trial 2	135.6	3.0	3.25
Trial 3	81.8	1.4	2.25

The time to complete a valve turn decreased as more compliant jogger features were included, and we saw a 2x improvement when both compliance and input space reduction was utilized. Additionally, manipulator safety faults were triggered significantly fewer times, and subjective opinions of difficulty decreased with the full compliant jogger. These results clearly show the potential to complete complex contact tasks quicker, safer, and more easily.

V. CONCLUSIONS

In this work, we showed how to complete contact tasks that typically require high precision if performed quickly, easily, and without damage to the robot or environment. We introduce a jogger and selectively reduce the user's input space to reduce their burden to manage task forces and precision. When used in conjunction, these features reduced both time and errors when completing the task. We defined the necessary impedance and motion parameters for the controller as Affordance Primitives, which can be intuited from existing Affordance Templates to quickly complete undefined tasks using teleoperation.

From this, we anticipate that new ATs can be generated automatically "behind the scenes" from Affordance Primitives used during successful teleoperation tasks. Furthermore, we recognize that as the set of available templates for a given ecosystem (particular robot in a particular environment) grows, there will exist a training set available to explore methods to identify Affordance Primitives for new tasks using existing machine learning methodologies, and potentially removing the developer from the effort of determining affordances altogether. Finally, we note the opportunity to extend and generalize affordances to perform complex contact tasks even involving multiple manipulators.

ACKNOWLEDGMENT

We would like to thank Woodside Energy for sponsoring the motivating work behind this project, and Dr. Andy Zelenak for his contribution to the compliant jogger.

REFERENCES

[1] R. C. Goertz, *Master-slave Manipulator*. Argonne National Laboratory, 1949.

[2] R. Goertz, "Mechanical master-slave manipulator," *Nucleonics (U.S.) Ceased publication*, vol. Vol: 12, No. 11, 11 1954.

[3] D. Kent, C. Saldanha, and S. Chernova, "A comparison of remote robot teleoperation interfaces for general object manipulation," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 2017, pp. 371–379.

[4] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, 2012, pp. 1–8.

[5] H. Hedayati, M. Walker, and D. Szafir, "Improving collocated robot teleoperation with augmented reality," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 78–86.

[6] T. Chakraborti, S. Sreedharan, A. Kulkarni, and S. Kambhampati, "Alternative modes of interaction in proximal human-in-the-loop operation of robots," *arXiv preprint arXiv:1703.08930*, 2017.

[7] S. Hart, P. Dinh, and K. Hambuchen, "The affordance template ros package for robot task programming," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 6227–6234.

[8] H. A. Yanco, A. Norton, W. Ober, D. Shane, A. Skinner, and J. Vice, "Analysis of human-robot interaction at the darpa robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 3, pp. 420–444, 2015.

[9] J. Gibson, "The ecological approach to visual perception. boston, ma, us," 1979.

[10] I. A. Sucan and S. Chitta, "MoveIt!" "http://moveit.ros.org/", 2017, accessed: 2017-02-14.

[11] S. J. Jorgensen *et al.*, "Deploying the nasa valkyrie humanoid for ied response: An initial approach and evaluation summary," in *19th IEEE international conference on humanoid robots*. IEEE, 2019.

[12] A. Zelenak. A reactive jogger for teleoperation and contact tasks. ROSCon. https://roscon.ros.org/2019/talks/roscon2019_jog_arm_final.pdf. [Online]. Available: https://vimeo.com/378683001

[13] M. Fallon, S. Kuindersma, S. Karumanchi, M. Antone, T. Schneider, H. Dai, C. P. D'Arpino, R. Deits, M. DiCicco, D. Fourie, *et al.*, "Affordance-based perception and whole-body planning in the darpa robotics challenge," *Tech. Rep. MIT-CSAIL-TR-2014–003*, 2014.

[14] T. Koolen *et al.*, "Summary of team ihmc's virtual robotics challenge entry," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2013, pp. 307–314.

[15] P. Beeson, S. Hart, and S. Gee, "Cartesian motion planning & task programming with craftsman," in *Robotics: Science and Systems Workshop on Task and Motion Planning*, 2016.

[16] S. Hart, P. Dinh, and K. A. Hambuchen, "Affordance templates for shared robot control," in *2014 AAAI Fall Symposium Series*, 2014.

[17] J. James, Y. Weng, S. Hart, P. Beeson, and R. Burridge, "Prophetic goal-space planning for human-in-the-loop mobile manipulation," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 1185–1192.

[18] P. Beeson and B. Ames, "Trac-ik: An open-source library for improved solving of generic inverse kinematics," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 928–935.

[19] T. McMahon, O. C. Jenkins, and N. Amato, "Affordance wayfields for task and motion planning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2955–2962.

[20] N. Yamanobe, W. Wan, I. G. Ramirez-Alpizar, D. Petit, T. Tsuji, S. Akizuki, M. Hashimoto, K. Nagata, and K. Harada, "A brief review of affordance in robotic manipulation research," *Advanced Robotics*, vol. 31, no. 19-20, pp. 1086–1101, 2017.

[21] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[22] H. Seraji, "Adaptive admittance control: An approach to explicit force control in compliant motion," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE, 1994, pp. 2705–2712.

[23] C. Schindlbeck and S. Haddadin, "Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 440–447.

[24] I. D. Walker, "The use of kinematic redundancy in reducing impact and contact effects in manipulation," in *Proceedings., IEEE International Conference on Robotics and Automation*. IEEE, 1990, pp. 434–439.

[25] M. H. Raibert and J. J. Craig, "Hybrid Position/Force Control of Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 126–133, 06 1981.

[26] H. Zhang and R. Paul, "Hybrid control of robot manipulators," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1985, pp. 602–607.

[27] N. Hogan, "Impedance Control: An Approach to Manipulation: Part I—Theory," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 1–7, 03 1985.

[28] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," *IEEE Journal of Robotics and Automation*, vol. 17, no. 1-19, p. 16, 2004.

[29] G. Golub and W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix," *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, vol. 2, no. 2, pp. 205–224, 1965.

[30] R. Valner, K. Kruusamäe, and M. Pryor, "Temoto: Intuitive multi-range telerobotic system with natural gestural and verbal instruction interface," *Robotics*, vol. 7, no. 1, 2018. [Online]. Available: https://www.mdpi.com/2218-6581/7/1/9