Improving Robustness in Complex Tasks for a Supervisor Operated Humanoid

Benzun P. Wisely Babu[1] Ruixiang Du[2] Taskin Padir[3] and Michael A. Gennert[4]

***Abstract*— Complex manipulation tasks in uncontrolled environments are challenged with errors from multiple sources that can prevent successful completion. We describe a framework for the decomposition of a complex task into behaviors for a supervisor controlled robot. A classification of behaviors based on the actors and dominant motion is used to analyze the success rate. Three methods to improve robustness are presented: reduction in length of robot-environment manipulation by using robot-only prepositioning behaviors, behavior definition using environmental constraints, and supervisor fine tuning during sub-task switching. We show the application of this framework for the wall task in the DARPA Robotics Challenge. The framework produces a robust successful implementation of the wall task with a duration of less than 10 min.**

I. INTRODUCTION

Humanoids are designed to perform actions in an environment designed for humans. However, performing complex manipulation tasks in an uncontrolled environment still poses challenges. The difficulty arises from numerous sources of errors such as sensor noise, modelling errors, tracking error etc. A complex task can be decomposed into behaviors that are robust enough to overcome these errors. In this paper we classify the behaviors to analyze the errors associated with them and present methods to compose the behaviors in a robust manner.

Composing complex tasks using object affordance has been employed by multiple groups [1][2][3]. Such approaches rely heavily on the perception of objects in the environment. These approaches do not reason about the generation of robot behavior sequences and the errors associated with them. We follow a decomposition of complex tasks into a behavior sequence while not directly attaching the behaviors to an object.

A complex task is divided into sub tasks composed of behaviors. Behaviors are classified based on actors involved as robot-only, robot-object, robot-environment and based on dominant actions as manipulation, mobility and perception behaviors. We analyze the robustness of a sequence by identifying the different errors associated with the behaviors. We find robot-environment manipulation behaviors to have more sources of errors and thus try to reduce reliance on such behaviors.

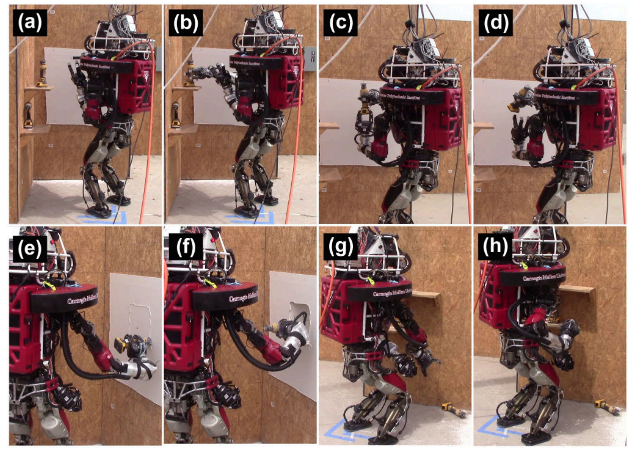

Fig. 1: Wall Task Sequence: a. Walk to drill with prep-pose; b. Grab drill; c. Switch on drill; d. Check drill state; e. Cut wall; f. Punch wall to remove piece; g. Drop drill; h. Recover pose and proceed to next task

We propose reducing the length of robot-environment manipulation behaviors by using robot-only prepositioning manipulation behaviors. We also observe the improvement in robustness of a behavior by using environmental constraints. Finally supervisor intervention by fine adjustment between sub task transitions can be used to ensure that the errors in sub-task preconditions are bounded.

The DARPA Robotics Challenge(DRC) introduced a set of difficult walking and manipulation tasks to evaluate a robot's ability to perform in a real life post-disaster environment with limited communication bandwidth. The challenge required the completion of complex manipulation task by a robot-supervisor pair in limited communication. We applied our approach to the wall task of the DRC.

The main contributions of this paper are as follows:

- A framework for decomposition of a complex task into behaviors and classification of behaviors based on actors involved and the dominant action performed by the robot.
- Strategies to improve success and robustness using prepositioning robot-only behaviors, environmental constraints and supervisor fine adjustments.
- Successful implementation of the framework on the Humanoid robot (ATLAS) for the wall task at the DARPA Robotics Challenge (fig.1).

[1]Benzun P. Wisely Babu is a PhD student, Robotics Engineering, Worcester Polytechnic Institute Worcester, MA 01609, USA bpwiselybabu@wpi.edu

[2]Ruixiang Du is a PhD student, Robotics Engineering, Worcester Polytechnic Institute Worcester, MA 01609, USA rdu@wpi.edu

[3]Taskin Padir is Asst. Professor at Robotics Engineering, Worcester Polytechnic Institute Worcester, MA 01609, USA tpadir@wpi.edu

[4]Michael A. Gennert is Director of Robotics Engineering, Worcester Polytechnic Institute Worcester, MA 01609, USA michaelg@wpi.edu

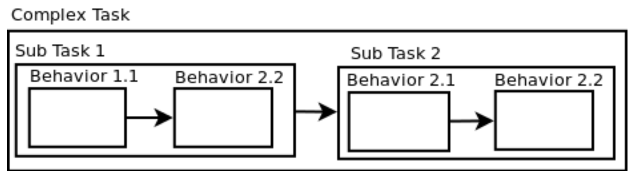

Fig. 2: Decomposition of a complex task into behaviors.

The following section gives a brief overview of behavior based approaches for complex tasks in humanoids. Section III presents a framework for decomposing complex tasks and improving robustness. It is followed by section IV which discusses the challenges in the DRC wall task. The software implementation for the task is presented in section V. We present our experiments and results section VI. Finally we present our conclusions and future work in section VII.

II. BACKGROUND

[3] discuss an affordance based architecture for planning in a complex environment for humanoids. [2] extend the approach to affordance template where they create an object template based on the object affordance and object action complex. [1] also employ an affordanced based approach for shared autonomy. The affordance based approach requires an object template to be associated for sets of actions. Since actions are associated with the object, there is a need to accurately locate and identify the object. The error in object identification can impose uncertainties in task completion. A complex task can also be decomposed based on task graphs[4].

There has been work in robot manipulation for reducing uncertainty in the system. [5] discuss a method to reduce pose uncertainty in manipulation planning by using task space regions. [6] discusses reducing uncertainty in tasks specified in an object frame.

[7] discusses the use of DRC Hubo for cutting a cardboard wall but they do not discuss task decomposition. [8] introduced how the wall task was done in the DRC Trails by team WRECS, the former name of team WPI-CMU.

III. DECOMPOSITION OF COMPLEX TASKS

A complex task can be decomposed into sub-tasks and behaviors as shown in Fig. 2. Behaviors are functional units that implement a controller to perform an action on the robot. In the context of humanoids the behaviors can be further classified based on the actors and the dominant motion involved.

A. Actor based classification

The three primary actors are the robot, environment and the object of interest. The actor based classification is as follows:

1) Robot only (R): The robot is a primary actor that is involved in all the behaviors. In these behaviors actions are performed without having to consider other actors in the world. These are behaviors where robot actions are limited by collision between the robot links. The error in the system is mostly dominated by errors from the internal state estimator of the system. For example, consider the action of lifting the hands of the robot in a open environment.

2) Robot - Environment (RE): The environment is composed of all the static and dynamic elements in the world. The elements can be wall, chair, table, other autonomous systems etc. The robot needs to take into account the collision and the state of objects before executing any action. For example consider a robot navigating through cluttered obstacles.

3) Robot - Object of Interest (RO): The objects of interest are elements in the environment with respect to which a behavior is defined. The robot needs to interact with these objects and will have to change the state of these objects. This is similar to the affordance based behavior models for robots. For example, consider in-hand rotation of a pencil by a robot.

B. Dominant action based classification

For robots to be capable of complex tasks, they have to execute three fundamental motions: Manipulation, Mobility and Perception.

1) Manipulation dominated behaviors: In Manipulation dominated behaviors the robot uses its end effectors to change the state of the world, such as pressing a button.

2) Mobility dominated behaviors: In mobility dominated behaviors the entire robot moves from one position to another in the world, such as moving from a kitchen to hall.

3) Perception dominated behaviors: In perception dominated behaviors the robot is making observations of the world, such as identifying a pen.

A behavior can be any combination of the actions and actors described above. The actor based classification and Dominant action based classification are independent classifications. Hence it is possible to describe a robot behaviors using a combination of both. For example a robot-environment manipulation task would be a robot lifting its hand up in a narrow corridor. In this case the robot will have to plan to avoid the environment while moving its end effector which is a manipulation task.

C. Robustness of behaviors

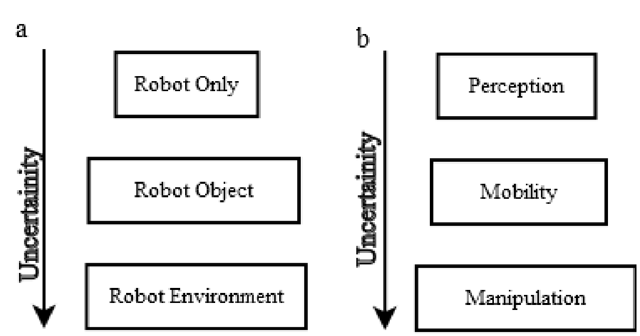

Fig. 3: The trend in uncertainty for the two types of classification of behaviors.

Errors and uncertainties are introduced at different levels due to incomplete and inaccurate perception of both the environment and the robot state. If the errors accumulate they can cause unsuccessful or wrong transitions between sub-tasks or improper evaluation of the preconditions for the sub-tasks.

It is possible to rank the behaviors based on the number of sources of error. Since a large amount of error in the system would mean more chances of failure and less robustness this would enable us to identify critical behaviors and implement schemes to improve their success. Fig. 3a and Fig. 3b show trend in robustness for actor based behavior and dominant motion based classification respectively.

The sensor noise and modelling inaccuracies are primary sources of error when we try to estimate the state of the world. The net error in the system increases when the number of independent elements that are needed to be tracked and modelled. For actor based system, the robot-environment case has the maximum number of objects involved, ie. robot, objects in the environment such as wall, table etc. This justifies the large uncertainty involved in such behaviors.

In a robot only perception, we are only trying to do internal state estimation of the robot. There is less error associated with it as it depends on values that the robot can directly observe. In robot-object case, we consider cases where the robot interacts with a single target object and will not be affected by other objects in the world. This includes in-hand manipulation such as switching an object from one hand to the other etc.

In dominant motion based classification, during perception behaviors only exteroceptive sensors are required. Since there exists multiple modalities of exteroceptive sensor i.e. such as laser and stereo the error in detection and tracking can be reduced given sufficient time. For manipulation and mobility based behaviors interoceptive sensors play a more dominant role. We have observed that the joint encoders on arms are more noisy and have more modelling errors than the ones on legs. So manipulation behaviors are less reliable than mobility behaviors.

D. Improving robustness

We describe three schemes we have successfully employed to improve the robustness of the behaviors.

1) Length of robot-environment behaviors: Since the estimation errors accumulate with the length of a behavior, reducing the length of a behavior improves robustness. It is possible to precede a long robot-environment behavior with a robot only behavior which will improve the reliability of the robot-environment behavior. For example a robot that needs to move through a narrow passage can preposition its end-effectors with minimal contact before entering the passage. The prepositioning is a robot-only behavior that can help reduce the elements in the robot environment that needs to be modelled and tracked.

2) Robust definition of a behavior: There sometimes exists ambiguities in the definition of a behavior. The ambiguities give rise to poor models of the environment and

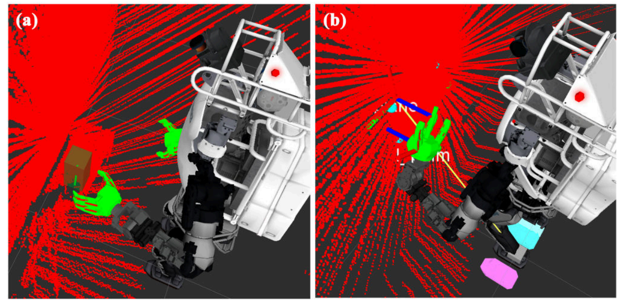

Fig. 4: (a) The hand orientation for cutting is constrained by the plane orientation. (b) The hand orientation for grabbing upright drill is constrained by the ground orientation.

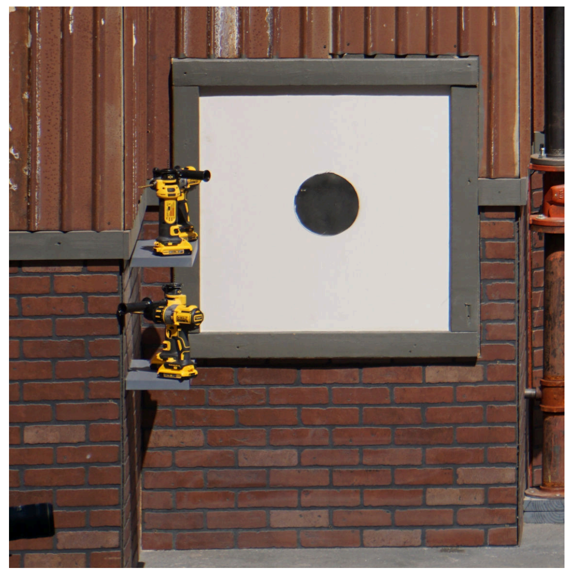

Fig. 5: Wall Task Configuration: The drills are placed on a shelf on the wall. The black circular area needs to be removed with the power tools.

the object of interest. This greatly reduces the reliability of the behavior. We can improve robustness of the behavior by adding additional constraints on it based on elements that are easily observed. For example the hand orientation for grabbing an upright drill can be matched with the normal to the floor. Also information from multiple sensors with different modalities can be used to reduce the uncertainty. Such as using the force torque sensors on the hand to further constrain the wall cutting (fig.4).

3) Supervisor intervention: Within a sub-task impacts of accumulating error become significant during transition between sub-task. As transitions are based on the belief that the preconditions are met, the accumulated error might prevent the continuation of the task. It is possible to involve the supervisor here to confirm and perform fine adjustment in-order to meet the preconditions more accurately. This creates robust transition from one sub-task to the next sub-task and improve the overall task success.

IV. DRC WALL TASK

In the DRC, the wall task simulates -disaster scenario in which a sector of wall needs to be removed so that the valve behind it can be reached and turned off. The intention of the wall task is to demonstrate the capability of a robot to operate power tools designed for humans to do heavy manipulation. The task requires the complete removal of a specified area (as shown in Fig. 5). The power tools are placed on shelves at different heights, one of which is approximately 32 inches above the ground and the other 44 inches. The drills provided for this task consist of two types, a cordless rotary cut tool and a pistol grip drill with optional side handle. The pistol drill has a bigger power button, and uses a larger drill bit at a lower RPM, which means it can be relatively easier to switch it on but harder to use it for cutting in the wall. The robot can pick up and use any of the four available drills during the competition. The wall to be cut is made up of sheet rock and a black circle with an approximate 8-inch (20 cm) diameter is drawn at the center.

In addition to the above requirements, there are also time constraints to this task. For the whole competition, the robot only has 60 minutes to finish all 8 tasks, which means in average only 7.5 minutes can be allocated for each task. In addition, the power drill can only run for about 5 minutes after it's switched on and has to be re-triggered afterwards. The robot has to finish cutting within 5 minutes after it turns on the drill.

A. Wall task decomposition

The wall task is divided into three manipulation sub-tasks as described below. Inspecting each sub-task further, we observe that all of them require coordinated behaviors for successful completion. These include perception behaviors to identify the environment and manipulation behaviors to interact with tools and wall. Since the robot doesn't start right in front of the wall task course during the competition, the robot also needs to perform mobility behaviors to ensure that the tools and the wall are within its reachable workspace. The wall task is decomposed as follows.

1) The behaviors of "pick up drill":
    a) preposition for grasping (Robot-only manipulation behavior)
    b) walk to wall task (Robot-only mobility behavior)
    c) detect drill and extract model features (Robot-only perception behavior)
    d) hand approaches drill for grasp (Robot-environment manipulation behavior)
    e) supervisor fine adjustment (Robot-supervisor perception behavior)
    f) grab drill in hand & pull drill out from the shelf (Robot-only manipulation behavior)
2) The behaviors of "switch on drill":
    a) move to a pre-defined two-handed posture (fig. 6) (Robot-only manipulation behavior)
    b) lose the grip on the drill hand (Robot-only manipulation behavior)

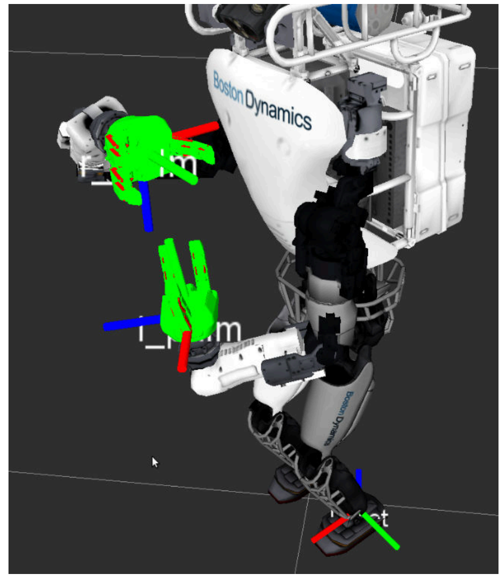

Fig. 6: The posture that the robot is in for switching on behavior. It is a manipulation robot only behavior.

    c) align other hand and grasp bottom of drill (Robot-object manipulation behavior)
    d) rotate drill with bottom hand and trigger power button with top hand (Robot-object manipulation behavior)
    e) operator check drill on/off (Robot-supervisor perception behavior)
    f) return to walk posture (Robot-only manipulation behavior)
3) The behaviors of "cut opening on the wall":
    a) operator specify cutting area(Supervisor-only perception behavior)
    b) raise hand with drill to a cutting preposition(Robot-only manipulation behavior)
    c) move hand to the starting point and push drill bit into wall (Robot-object manipulation behavior)
    d) cut a rectangle on the wall (Robot-object manipulation behavior)
    e) set hand to a waiting pose and wait for confirmation from operator (Supervisor-only perception behavior)
    f) punch wall to remove the cutting piece (Robot-object manipulation behavior)
    g) drop drill in front of robot (Robot-object manipulation behavior)

B. Robustness analysis

For successful completion of the wall task, the three subtasks need to be completed in sequence. This process requires the robot to perform all behaviors with high robustness and reliability. Even one transition error between two behaviors can lead to failure of the whole task. The time constraints make this problem even harder as they urge the robot to think and act faster. Thus to increase the success rate, we should analyze the sources of error from all the behaviors

listed above and find methods to reduce or eliminate those errors.

Strictly speaking, all behaviors will introduce modelling errors. But some of them can be easily controlled to have minimal impacts on the overall task performance while others may not. Take behavior 1a) as an example. In this step, the robot raises its arms to a pre-defined pose with less chance of environmental collision and walks to the wall task holding the posture. Since the robot starts from a relatively open space where wall and shelves are far away, it's safe to perform the arm pose change. By having the operator specify desired standing location, we can be confident that the robot is walking to the right target. Even if there exist small errors in the target location measurement and robot locomotion, they have very little negative effects on the subsequent manipulation tasks.

For most of the other manipulation dominated behaviors, errors are mainly introduced by perception and kinematic modelling uncertainities. For example, behavior 1c) is about moving the robot hand from its initial pose to the pose for grasping the drill. This process requires an online motion planning so that the robot can get a collision-free path in-between the two poses. However limited by perception precision, sometimes the shelves are not fully detected and represented in the planning scene. Kinematic tracking error also increases the possibility of collision since the operational space is very tight. As a result, the robot may hit the shelves from the bottom before it reaches the destination. If the collision is hard, it may cause the robot to lose balance and fall over. Moreover, even if the robot successfully avoids collision with wall and shelves, the hand may push the drill over when it tries to get close to the drill. To make this behavior more robust, a pre-defined pose is added in behavior 1a). That is raising the hand at a position higher than the shelves before it gets close to the task site. By doing so the shelves are not an obstacle between the two poses. In addition, we increased the distance between the final hand position and the drill position. To ensure the grasp quality, we bring in operator intervention. The operator adjusts the hand position based on visual feedback and confirms that the robot can grasp the drill.

Similar analyses are performed on all behaviors and these behaviors are adjusted if there are large number of error sources. When necessary a behavior is divided into a few sub-behaviors and robot-only behaviors or operator involvement are added before or after to reduce risk or inaccuracy.

V. SOFTWARE IMPLEMENTATION

For the DRC Finals, the robot works at the competition site and the operators monitor and control the robot remotely in the garage. There is a controlled communication link between the robot and the control station. Fig. 7 shows the structure of the software implementation for wall task to work in this configuration. Accordingly there are two parts in the software: one part runs on the robot and the other part runs on operator control units (OCUs).

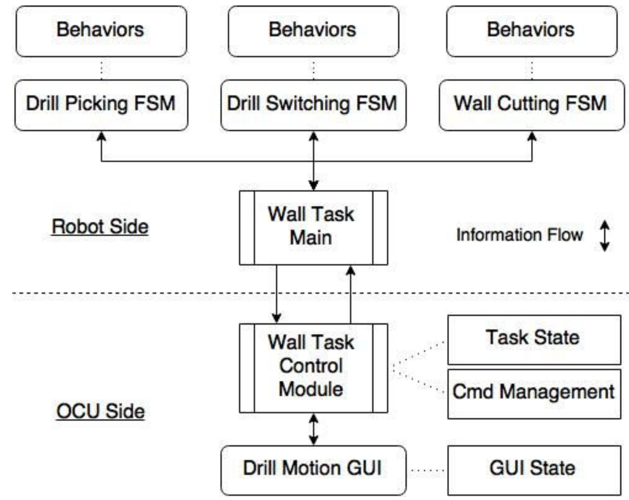

Fig. 7: Software Structure of Wall Task

On the robot side a hierarchical finite state machine (FSM) is implemented to control all the sub-tasks and the behaviors of each sub-task, which were discussed in Section IV. As shown in Fig. 7, the three state machines are drill picking FSM, drill switching FSM and wall cutting FSM. A "wall task main" node works as a higher-level FSM. It coordinates the transition of sub-tasks, receives commands from the OCU and sends feedback to the remote control station.

On the OCU side, a "wall task control module" runs in the background, above which the drill task GUI is shown. The control modules maintains the information exchange with the drill task state machine on the robot. It records the task state and updates the information to the user interface. At the same time it receives commands from the operator through GUI to control the remote state machine. This design ensures that the operator doesn't lose the task information even if the graphical interface crashes.

Fig. 8 shows the GUI we used in the DRC Finals. The GUI displays an image stream from the robot camera. The operator can mark (with a line or scribble) on the 2-D image to specify the area of interest, for example by doing so the operator can tell the robot the approximate drill location. The perception algorithm performs a quick statistical estimation using the laser and stereo points around the area of interest to generate the final target point for grasping and for cutting the wall, the user specifies the center and the radius of the black circle using two points. Then internally four connecting points that describe the rectangle for cutting around the circle are generated. With this rectangular cutting trajectory, the robot can automatically execute the cutting process. For behavior transitions, in most cases the operator just needs to provide a confirmation signal by clicking the "Next" button. In addition, wild card features are provided to give the operator freedom to jump to a specific behavior when the robot is not acting as expected.

Other considerations in the implementation include the robot balance control and full-body motion planning. To support the behaviors at the task level, we used the balancing

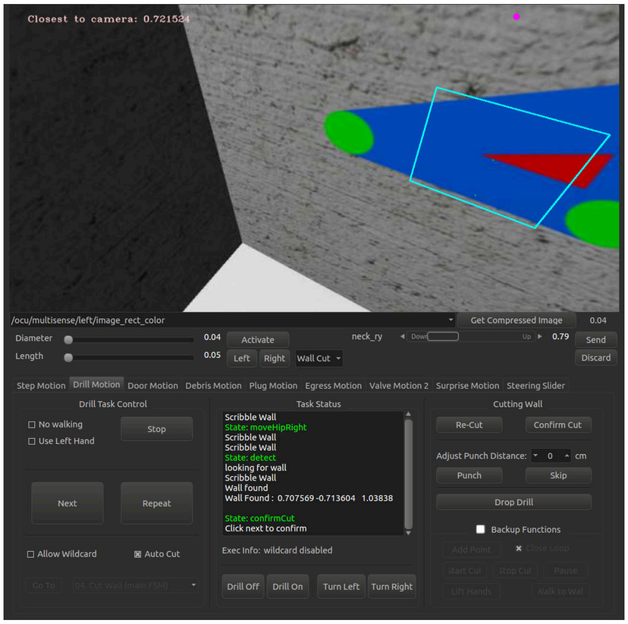

Fig. 8: The interface for the wall task. The area to be cut is marked with a rectangle on the image.

Fig. 9: Left picture: the different drill models used in DRC Trials and Finals respectively; Right picture: the knob installed on hand to press button of drill

control scheme discussed in [9], which controls the robot at joint level. For manipulation motion planning, we used TrajOpt [10] to generate collision-free trajectories.

VI. EXPERIMENTS & RESULTS

To get reliable and repeatable behaviors for the completion of the wall task, various experiments were conducted both in simulation and on the real Atlas robot. From experiments, we can better define a single behavior and find the transition conditions between two consecutive actions. For example, the position of the robot with maximum reachability in the area of the wall task course was found using repeated trial and error experiments. We also use experiments on the robot to figure out the strategy for a sub-task. Among the sub-tasks and behaviors listed in Section IV, the drill switching on sub-task and wall cutting behavior were the most difficult ones to deal with. We spent a large portion of the development time to solve these two problems.

In the DRC Trials, we used a ”shaking” method to switch the drill on [8]. By adding a hard knob on the robot finger and

TABLE I: Switch on sub-task completion time for the different starting conditions and approaches.

Approach	Drill starting orientation (deg)	Completion time (sec)
Two Handed	0	111
Two Handed	90	127
Two Handed	180	110
Two Handed	270	120

shaking the drill at a specific orientation, the power switch can be triggered on by the knob. However the performance of this behavior is not very predictable. It is based on the mass distribution of the given drill. Most of the time this behavior works well, but if the switch is not triggered properly on some occasions we don’t have much control to correct it. This violates our objective of improving robustness and hence we decided to look for a better solution. From Fig. 9 we can see that to switch the drill on, the robot has to press exactly on the small button from the right side (pressing from the other side is to switch the drill off). Considering the time constraint and the complexity of implementation, we decided not to use the strategy which uses one hand to hold the drill and the other hand to press the button directly. Instead, we tried to take advantage of our knowledge in the geometrical characteristics of the drill as what we did in the DRC Trials. Noticing that the middle part of the drill is thinner than two ends, the final strategy we decided to use is described as follows:

1) Hold the drill in the middle part when grabbing it.
2) Lift the drill in front of chest and keep the drill oriented vertically.
3) Loosen the hand a little but enough for the drill to drop until the edge between the upper end and the middle part gets stopped by the hand. Because the drill shape never changes and it always stops at the edge, we can get the drill at the same relative height with respect to the hand consistently.
4) We installed and adjusted a small knob on the hand (see Fig. 9) to make it at the same height of the drill button. We now need to rotate the drill so the switch will be turned on when the hand closes.
5) Use the other hand to hold the drill from the bottom. Once the hand holds the drill, we will know the final hand pose which ensures the drill button to be aligned with the closing finger.
6) Open and close the finger several times to trigger the button to switch the drill on.
7) Tilt the drill and check if the LED on the drill is on, which indicates the drill bit is rotating

Before we adopted this approach, we tried a few other variations. One strategy we tested was to control the hand to trigger the button when the hand approaches the drill and closes the hand for grasping. However this strategy requires the accurate control of the hand height and orientation. If the drill is not placed on the table with a good orientation for the grasp, the robot may need to rotate the drill first before

TABLE II: Time analysis of the final implementation

#	Grab Drill (sec)			Switch On (sec)			Cut Wall (sec)		
	preposition	pick up	pull out	preposition	alignment	confirmation	preposition	cutting	punch and drop
1	14	56	12	16	74	28	28	68	56
2	12	60	13	17	68	30	29	70	52
3	13	54	14	16	75	24	27	65	52

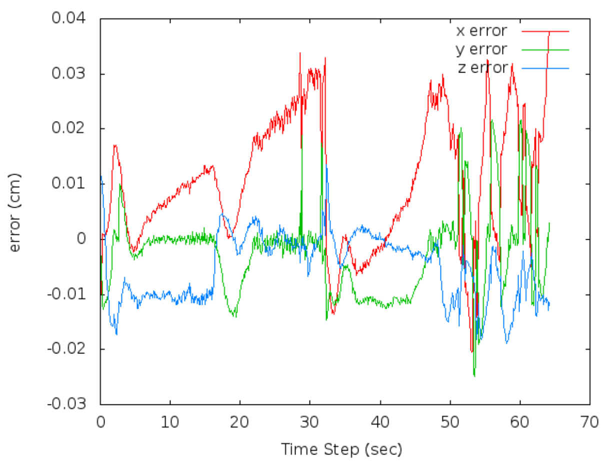

Fig. 10: The error in cutting.

attempts to grasp and trigger on. The rotation motions can sometimes take a long time. Comparatively the strategy we finally use is very deterministic and less time consuming (table. I) no matter how the drill is placed on the table.

For the wall cutting process, we first only implemented position control and controlled the hand to cut by following a rectangular trajectory in the wall plane. The problem we observed from the experiments was that the drill sometimes may push the wall too hard or comes off from the wall due to poor trajectory tracking. This behavior may lead to bad cutting quality or even push the robot over. To make sure the drill bit always penetrates into the wall at the desired depth, we added an extra force control loop along the normal direction of the wall (Fig. 10). The combined position and force control guarantee the cutting process is safe and of high quality.

Though more than 20 behaviors need to be executed in the right order for the wall task, the whole process is very robust and repeatable with our task decomposition and software implementation. We refine the boundary of each behavior and add robot-only behaviors to get certain pre-conditions to reduce uncertainly. For behaviors with large uncertainties, we effectively controlled the possibility of failure by introducing human supervisor. The Wall task approximately took 8-10 minutes. A time analysis of the different sub-task is presented in table. II.

A. DRC Finals

Unfortunately, we were unable to complete the wall task during the DRC Finals. On the first day, the robot misjudged a state transition condition. Instead of proceeding to switch on the drill, it attempted to adjust its standing position and retry grasping the drill, while the drill was already successfully grasped in the hand. This lead to the robot dropping the drill and we had to abort this task. On the second day, the robot successfully grabbed the drill and switched it on. But when robot was about to start cutting in the wall, it encountered a forearm mechanical failure. One joint in the forearm stopped working and it became impossible to continue the force and position control of the arm. We successfully prevented the robot from falling but had to abort the task again. The first failure could have been avoided if we had tested the state machine implementation more thoroughly. The misjudged transition condition could have been defined to be more robust to avoid wrong judgement. As to the second failure, the main cause is that we ignored the robustness of the robot hardware. Although the two-arm strategy gave us a higher success rate than a one-arm strategy in switching on the drill, the pre-condition of the two-arm strategy often cannot be met; both robot forearms need to work properly. If we could develop a one-hand strategy as a backup, we would be able to deal with this kind of unpredictable mechanical failure in a better way. However, the competition proves that our decision on using force control during cutting is correct. Several teams failed because they did not drill deeply enough. MIT's drill actually came out on day 2, causing them to fail to do the task. A more thorough discussion of this and other ATLAS shortcomings may be found in [11].

VII. CONCLUSIONS

We have presented an architecture for decomposing complex tasks into behaviors. We have shown methodologies to reduce uncertainties and improve robustness of the robot task. Thought we couldn't finish the wall task during the competition, our overall success rate for the task was high during practice. We believe that the architecture can be extended to other robots and tasks.

ACKNOWLEDGMENT

The authors would like to thank the other members of team WPI-CMU for their assistance conducting experiments. This work is sponsored by Defense Advanced Research Project Agency, DARPA Robotics Challenge Program under Contract No. HR0011-14-C-0011. We also acknowledge our corporate sponsors NVIDIA and Axis Communications for providing equipment support.

REFERENCES

[1] N. A. Radford, P. Strawser, K. Hambuchen, J. S. Mehling, W. K. Verdeyen, A. S. Donnan, J. Holley, J. Sanchez, V. Nguyen, L. Bridgwater, R. Berka,

R. Ambrose, M. Myles Markee, N. J. Fraser-Chanpong, C. McQuin, J. D. Yamokoski, S. Hart, R. Guo, A. Parsons, B. Wightman, P. Dinh, B. Ames, C. Blakely, C. Edmondson, B. Sommers, R. Rea, C. Tobler, H. Bibby, B. Howard, L. Niu, A. Lee, M. Conover, L. Truong, R. Reed, D. Chesney, R. Platt, G. Johnson, C.-L. Fok, N. Paine, L. Sentis, E. Cousineau, R. Sinnet, J. Lack, M. Powell, B. Morris, A. Ames, and J. Akinyode, "Valkyrie: Nasa's first bipedal humanoid robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, 2015. [Online]. Available: http://dx.doi.org/10.1002/rob.21560

[2] A. Romay, S. Kohlbrecher, D. Conner, A. Stumpf, and O. von Stryk, "Template-based manipulation in unstructured environments for supervised semi-autonomous humanoid robots," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, Nov 2014, pp. 979–986.

[3] M. Fallon, S. Kuindersma, S. Karumanchi, M. Antone, T. Schneider, H. Dai, C. P. D'Arpino, R. Deits, M. DiCicco, D. Fourie, T. Koolen, P. Marion, M. Posa, A. Valenzuela, K.-T. Yu, J. Shah, K. Iagnemma, R. Tedrake, and S. Teller, "An architecture for online affordance-based perception and whole-body planning," *Journal of Field Robotics*, vol. 32, no. 2, pp. 229–254, 2015. [Online]. Available: http://dx.doi.org/10.1002/rob.21546

[4] K. Okada, A. Haneda, H. Nakai, M. Inaba, and H. Inoue, "Environment manipulation planner for humanoid robots using task graph that generates action sequence," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 2, Sept 2004, pp. 1174–1179 vol.2.

[5] D. Berenson, S. S. Srinivasa, and J. J. Kuffner, "Addressing pose uncertainty in manipulation planning using task space regions," in *in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*, 2009.

[6] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decr, R. Smits, E. Aertbelin, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007. [Online]. Available: http://ijr.sagepub.com/content/26/5/433.abstract

[7] R. O'Flaherty, P. Vieira, M. Grey, P. Oh, A. Bobick, M. Egerstedt, and M. Stilman, "Humanoid robot teleoperation for tasks with power tools," in *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*, April 2013, pp. 1–6.

[8] M. DeDonato, V. Dimitrov, R. Du, R. Giovacchini, K. Knoedler, X. Long, F. Polido, M. A. Gennert, T. Padr, S. Feng, H. Moriguchi, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Human-in-the-loop control of a humanoid robot for disaster response: A report from the DARPA Robotics Challenge Trials," *Journal of Field Robotics*, vol. 32, no. 2, pp. 275–292, 2015. [Online]. Available: http://dx.doi.org/10.1002/rob.21567

[9] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization-based full body control for the DARPA Robotics Challenge," *Journal of Field Robotics*, vol. 32, no. 2, pp. 293–312, 2015. [Online]. Available: http://dx.doi.org/10.1002/rob.21559

[10] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization," in *in Proc. Robotics: Science and Systems*, 2013.

[11] C. G. Atkeson, B. P. W. Babu, N. Banerjee, D. Berenson, C. P. Bove, X. Cui, M. DeDonato, R. Du, S. Feng, P. Franklin, J. P. Graff, P. He, A. Jaeger, J. Kim, K. Knoedler, L. Li, C. Liu, X. Long, F. Polido, M. Gennert, T. Padir, G. G. Tighe, and X. Xinjilefu, "NO FALLS, NO RESETS: Reliable Humanoid Behavior in the DARPA Robotics Challenge," in *submission to Humanoid Robots (Humanoids), 2015 15th IEEE-RAS International Conference on*, 2015.