

Guided Manipulation Planning at the DARPA Robotics Challenge Trials

Christopher Dellin, Kyle Strabala, Galen Clark Haynes, David Stager, and
Siddhartha S. Srinivasa

National Robotics Engineering Center
The Robotics Institute, Carnegie Mellon University

Abstract. We document the empirical results from Carnegie Mellon University’s entry into the DARPA Robotics Challenge. Our system seamlessly and intelligently integrates recent advances in autonomous manipulation with the perspective and intuition of an expert human operator. Virtual fixtures are used as the common language between the operator and the motion planner. The planning system then solves a guided manipulation problem to perform disaster-response tasks.

1 Introduction

Motivation. Despite significant advancements in autonomous robotic manipulation, disasters such as the 2011 Fukushima Daiichi nuclear meltdown and the 2010 Deepwater Horizon oil spill have exposed severe limitations. Specifically, even when intact, these environments are engineered for human operation but not structured for robots. Worse, in a disaster, the environment changes unpredictably and significantly. Furthermore, errors in manipulation can be fatal not just for the robot, but can result in collateral damage.

When faced with such challenges, successful approaches often resort to direct teleoperation [16], as demonstrated in nuclear material handling [7], explosive ordnance disposal (EOD) [8], and assistive surgery [10]. Work to improve this interaction method has focused on providing assistance to the operator, e.g. graphical feedback on task orientations [5]. However, disaster scenarios have severe limitations in *bandwidth* and *latency*, making direct teleoperation challenging and time-consuming. We aim for a fast and robust approach to this problem which allows a robot directed by a human operator to efficiently perform task-relevant behaviors over a low-fidelity communications link.

Problem Statement. The DARPA Robotics Challenge (DRC) program is a competition which aims to “develop ground robots capable of executing complex tasks in dangerous, degraded, human-engineered environments”.¹ Carnegie Mellon’s National Robotics Engineering Center entered the competition, and developed the CHIMP robot² [14] and its software system that demonstrates significant functionality in these tasks.

¹ http://www.darpa.mil/Our_Work/TT0/Programs/DARPA_Robotics_Challenge.aspx

² Carnegie Mellon University Highly Intelligent Mobile Platform

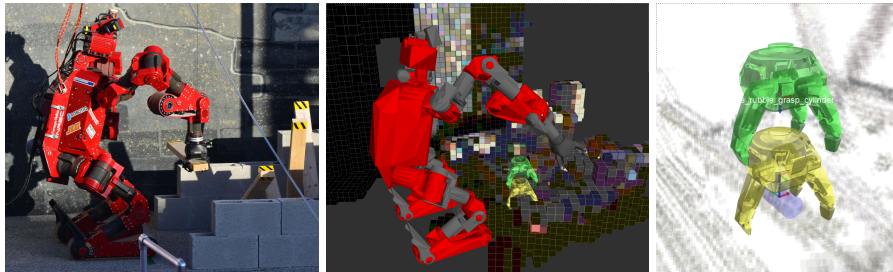


Fig. 1: CHIMP clearing ten pieces of wood during the Debris task at the DRC Trials, along with the operator’s reconstructed 3D environment containing fixtures which comprise the common language between the operator and the manipulation planner.

Contributions. Our key tenet is to efficiently integrate recent advances in autonomous manipulation (e.g. [1]) with the perspective and intuition of an expert human operator using *virtual fixtures* (Figure 1). Instead of their typical use as an aid to the operator during teleoperation [13], we use virtual fixtures as a *common language* that enables the operator to specify constraints and annotations to the robot, which it then addresses autonomously. We expect that the system’s manipulation capability under real-world task conditions will trigger further development in both theoretic research and system design.

Related Work. Our work is related to shared-autonomy manipulation planning systems such as the “human-in-the-loop” system developed by [9]; we extend their approach to general sets of constraints in a concurrent, multi-operator framework. We also note the relation to spacecraft telerobotics approaches (e.g. [17]) which must accommodate significant latency, albeit without the time pressure inherent in disaster response scenarios.

Outline. This paper presents an outline of our approach and analyzes our results from the DRC Trials, which took place in Homestead, FL in December 2013. We present our technical approach in Section 2. Section 3 details the application of the approach to each of the five manipulation tasks as the Trials, and Section 4 briefly describes our aggregated results and discusses experimental insights.

2 Technical Approach

A comprehensive solution to this problem requires a wide range of intelligent components (e.g. high-performance hardware, compressed world modeling, etc.). In this paper, we focus on the foundational components used for describing the five manipulation tasks between the operator and the CHIMP planning and execution system. Details of the full system are forthcoming in [14].

Complementary Skills. When developing our approach to the manipulation tasks at the challenge, we wanted the system to exploit the complementary skills of the operator and the robot; see discussions of these skills e.g. in surgery [15] or space robotics [12]. The operator is generally endowed with *task awareness*,

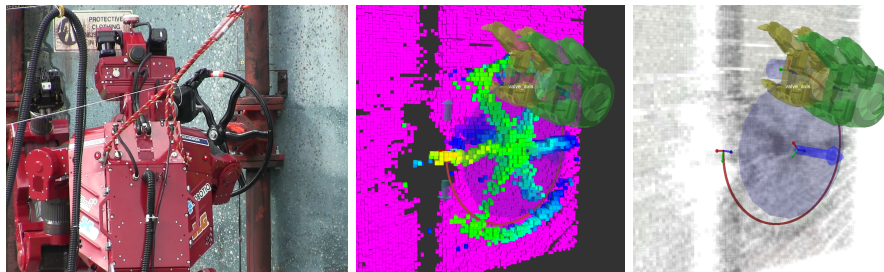


Fig. 2: CHIMP preparing to turn a valve at the DRC Trials, along with a *GraspCylinder* fixture (upper cylinder and grippers) and an *Axis* fixture (lower disk and axis arrow) configured for a 270 degree clockwise turn.

a high-level understanding of the task to be performed, along with the sequence of steps the robot must execute; however, direct teleoperation approaches often fail, especially under restricted networking fidelity or for constrained tasks. In contrast, the robot’s motion planners are equipped with *kinematic awareness*, the capability to search and optimize over complex kinematic, collision, and stability constraints, but fully autonomous task planners suffer from brittleness and overfitting, particularly in unstructured environments. An efficient design of the interface between the two is therefore essential.

Fixtures as a Common Language. We chose to use a variant of *virtual fixtures* as this primary interface. In a traditional telepresence system, a virtual fixture [13] serves as a perceptual overlay or constraint, introduced to an operator’s understanding of the remote scene, designed to reduce their requisite mental and sensory workload and improve task performance. Fixtures have been applied to fields such as medical [4] and space [17] robotics.

While our fixtures do serve this purpose for the operator, we also focus on the dual purpose: using fixtures as task-relevant specifications for the robot’s planning and execution system. Fixtures are first-class objects, which can be defined by the operator in order to build a context for the robot to perform a task, or placed/adjusted automatically by the robot in response to perception data or its internal world model. Fixtures are spatial entities that are defined in *task space* relative to a world frame, robot link, or another fixture. To the operator, fixtures are presented and manipulated by overlaying them onto the voxelized world model.

Our key insight is that virtual fixtures, as first-class spatial entities, are an effective interface between the operator and the robot. They enable each to exercise their strengths, allowing the operator task-level control while abstracting away the complexities of the robot’s kinematics, geometry, stability, etc. To the robot, fixtures impart task-rooted guidance, constraints, and intermediate goals which focus the planning problem.

Examples and Usage of Virtual Fixtures. In this paper, we focus primarily on the use of fixtures as task-space kinematic constraints. Each fixture *targets* either a robot link (e.g. gripper) or another fixture and advertises one or more

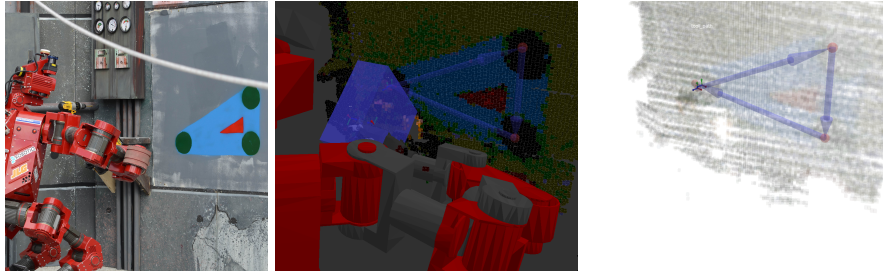


Fig. 3: CHIMP preparing to cut a triangular pattern from a wall in the DRC Trials, along with a `Box` fixture representing the volume of the drill in the hand, and a `PlanarPath` fixture showing the triangular shape to be cut.

named constraints for that target. Such constraints are then available to be selectively referenced by the planning and execution system, often in sequence for a particular task.

For example, the `Axis` fixture codifies a generic rotational constraint about a fixed axis (Figure 2). It is placed by the operator in the 3D world by first selecting a center point, and then selecting at least three points which define a plane normal to the axis. The fixture can be configured with labeled angles (e.g. `min/max`, `goal`, etc.). During the Door and Valve tasks, the handle was fixtured by a `GraspCylinder`, and the door/valve body by an `Axis` which targetted the `GraspCylinder` (i.e. constrained it to lie along a particular manifold).

The `GraspCylinder` fixture represents a graspable object using a simple grasping strategy that includes a configurable target gripper and grasp/pregrasp offsets. Figure 1 shows this fixture applied to a debris piece at the Trials. It advertises the `pregrasp`, `approach`, and `grasp` constraints, any of which may be active when the appropriate gripper satisfies the specification.

Fixtures may also be more task-specific when warranted; a `PlanarPath` fixture was created for the Wall task, which allowed the operator to define an arbitrary path of linear segments on a 2D surface (see Figure 3). In preparation for the Trials tasks, we created instances of several other fixtures, including the `Vector`, `Box`, and `RotaryCutTool` which will not be discussed in detail.

Guided Manipulation Planning. To the robot’s motion planning system, each fixture defining a Cartesian kinematic constraint induces a manifold in the robot’s configuration space. We chose to represent these constraints on target objects in the scene via Task Space Regions (TSRs) [3]. Since all fixtures are first-class objects available to the planning system, multi-step planning requests can simply reference fixture identifiers. For example, this composite request represents a valve turn:

```
Plan(to=GraspCylinder1.pregrasp)
Plan(along=GraspCylinder1.approach, to=GraspCylinder1.grasp)
Plan(along=Axis1.initial_to_goal, to=Axis1.goal)
Plan(along=GraspCylinder1.approach, to=GraspCylinder1.pregrasp)
```

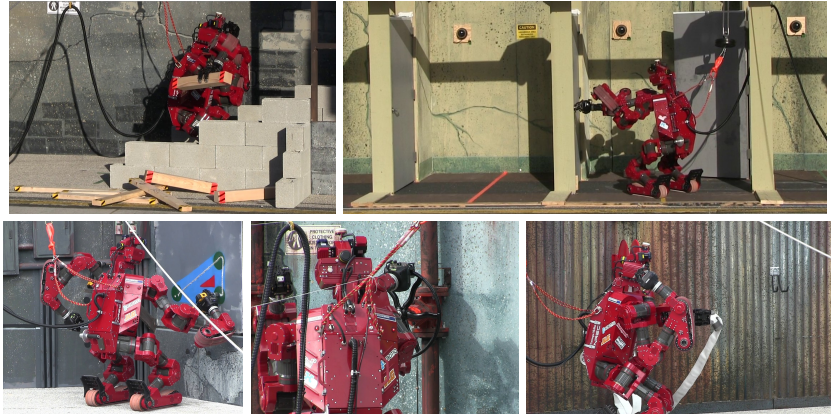



Fig. 4: CHIMP performing the five manipulation tasks at the DRC Trials: Debris, Door, Wall, Valve, and Hose.

In this way, the role of the planner is simply to move the system between and along the constraint manifolds induced by the fixtures. The robot is tasked with what we term the *guided manipulation problem*, finding feasible paths guided by the given ordering of constraints. While the request above may be composed manually by the operator, the system provides shortcuts via “wizards” for composing common tasks (e.g. pick-and-place, operating valves/hinges, and completing a wall cut). During the Trials, we used the CBiRRT algorithm [2]) for planning with configuration-space constraint manifolds.

Trajectory Confirmation and Supervision. Once a candidate plan is computed, the resulting trajectory can be previewed by the operator before it is sent to the robot. Once executing, it can be supervised by the operator, who may choose to step through the trajectory one segment at a time for close inspection.

Trajectory Execution and Control. The robot executor maintains a queue of trajectory segments to be executed. Each segment produced by the planner is *tagged* with the fixture(s) that were asserted to be active; this allows the executor to validate that the fixtures’ constraints are still met at the time of execution, and adjust the trajectory accordingly in some cases.

Trajectory segments which respect certain types of fixtures are tagged to be executed in particular ways; for example, segments which induce closed kinematic chains (e.g. valve turns) are executed using a workspace force controller which allows for looser gains in overconstrained directions.

3 Experimental Results

We designed a flexible system consisting of mobility actions, annotation tools, constrained motion planners, and tele-operation primitives that can be adapted to different task workflows. We leveraged prior work for the infrastructure (ROS [11]) and planning environment/algorithms (OpenRAVE [6], CBiRRT [2]), as

well as significant technologies developed at the NREC. Here, we detail CHIMP’s performance during the five manipulation tasks at the DRC Trials (see Figure 4).

Network. During the competition, operators were physically isolated from the task area, with the network link alternating every minute between 1Mbps bandwidth / 100ms latency and 100kbps bandwidth / 1s latency.

Scoring. Teams were allowed 30 minutes to complete each task, and were judged by three metrics prioritized to break ties:

1. Points (Task Completion): teams were awarded 3 points for full completion of each task. Partial points were awarded for defined subtask completion.
2. Interventions: teams were permitted to (but penalized for) manually intervening during the task execution (e.g. falling on the safety belay). A bonus point was awarded for each task in which no interventions were needed.
3. Completion Time: aggregated time taken by the robot to perform the tasks.

The remainder of this section describes in detail how our technical approach was applied to each of the five manipulation tasks, and provides an analysis of time spent. We only provide a cursory description of the tasks themselves; we invite the reader to review the full task descriptions and detailed rules for the Trials competition at the DARPA archive website.³

Multiple Operators. For each task at the Trials, between two and four operators interacted simultaneously with the robot and its internal hardware, perception, fixturing, planning, execution, and teleoperation systems. Because fixtures are first-class objects, they can be defined by one operator, but seen, modified, and referenced by plans from other operators.

Timeline Figures. For each task, a figure is provided which details the fixtures, plans, network statistics, and robot execution mode throughout the 1800s of allotted time (Figures 6 – 10).

Fixture lifetimes are shown first. Fixtures are shared between operators once they are first created and saved. Black activity bars denote times when an operator is actively modifying fixture parameters. Planning durations are shown next, along with connections to the fixtures referenced in each planning request. Only operators who adjusted fixtures or requested plans are shown in these figures.

Estimates of network latency and aggregated bandwidth from (“Rx”) and to (“Tx”) the robot are shown. The robot executor was modal, either in trajectory execution (“Trj”), end-effector teleoperation (“Tel”), or driving (“Drv”) mode; black activity bars denote individual trajectory segments or approximate motion request bandwidth. Last, estimates of torso speed and times of points awarded are illustrated.

³ <http://archive.darpa.mil/roboticschallengetrialsarchive/>

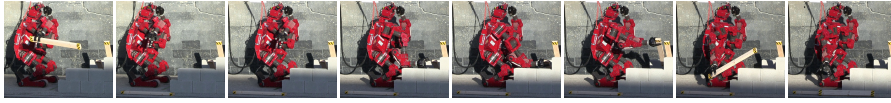


Fig. 5: During the Debris task, CHIMP clears piece 2, between 150s and 310s.

3.1 Debris Task

Debris Setup. Each robot was allowed to **Setup** behind a start line. The task involved removing five pieces of wooden debris (1 pt.), removing an additional five pieces of debris (1 pt.), and driving through the open doorway (1 pt.). The approximate configuration of the debris was provided apriori.

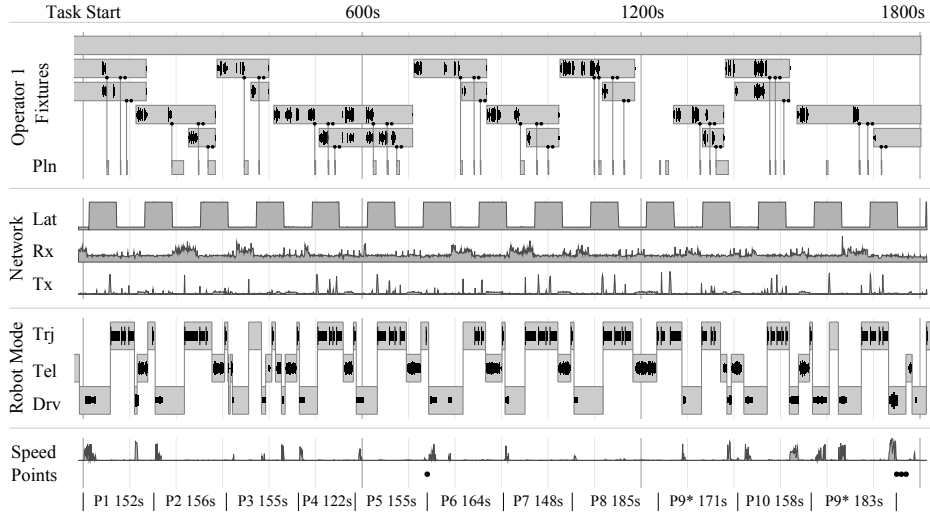
Debris Approach. Figure 6(b) outlines our approach for the debris task. The task was roughly organized into an initial **Setup** phase, followed by a four-step cycle of **Drive**, **Grasp**, **Liftoff**, and **Drop** phases. During the task, four operators operated the robot simultaneously, with loose roles of (a) world modeling, (b) fixturing/planning, (c) teleoperation, and (d) hardware monitoring.

The Debris task used the **GraspCylinder** fixture to define a grasp of each piece, along with a **Vector** fixture to define its liftoff direction and distance.

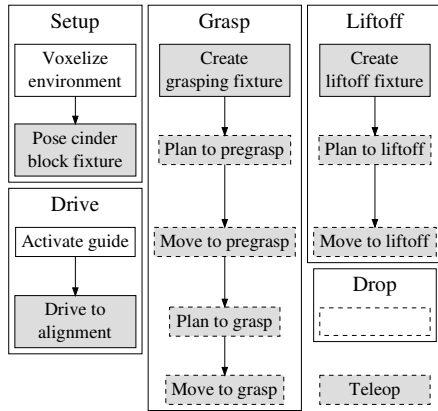
During **Setup**, the world modeler constructed approximate volumetric models of several static objects in the environment, in order to improve collision checking speed and accuracy. The **Grasp** and **Liftoff** phases are primarily managed by the fixturing/planning operator, who creates virtual fixture annotations for pregrasp, grasp, and piece liftoff constraints for the planner, invokes the planner, reviews the proposed trajectory, and supervises its execution. Once the piece is grasped and lifted, in the **Drop** phase, the third operator teleoperates the gripper towards a rough drop area to drop the piece.

Debris Results. We achieved four points in 1749s; see Figure 1 for a view of CHIMP mid-task. We prematurely dropped the ninth piece, and returned to it after completing the subsequent piece; the drop occurred at 1380s, during its teleoperation move to the drop zone. We took advantage of pipelining (see Figure 6(c)); for example, for many pieces, fixtures for subsequent pieces were created and positioned prior to the current piece being dropped.

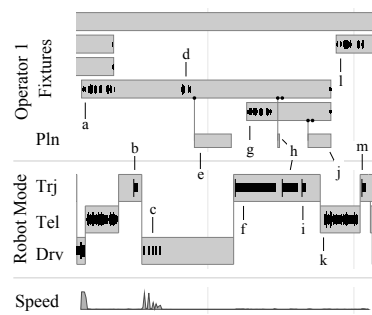
A log of data collected at the debris task is shown in Figure 6(a). We moved all ten pieces in approximately 29 minutes. Figure 6(c) shows an annotated log of the process to clear the second of the ten pieces, and Figure 5 shows the robot during its motion. The approach provides for several opportunities for pipelining. First, the fixture for the piece was created (a) before the previous piece was dropped (b), and before the robot was positioned appropriately (c). Second, the liftoff fixture was created and positioned (g) while the to-pregrasp motion was being executed. This piece also demonstrates a fail-over strategy that we used, whereby a long-running plan (j) or execution was interrupted and performed by teleoperation (k).



(a) Full debris task, moving ten pieces in 1749s. Piece 9 (*) was dropped during its first attempted transfer, and was later retrieved to complete the task.



(b) Flowchart of debris task. The cinder block fixture represents the environment; it is the first fixture created (top row in Fig 6(a)) and exists for the entire task. Dashed-border steps transition to teleoperation on failure.



- a Create new grasp fixture
- b Drop previous piece
- c Drive to alignment
- d Adjust grasp fixture
- e Plan to pregrasp
- f Execute plan to pregrasp
- g Create liftoff fixture
- h Plan and execute grasp
- i Close gripper
- j Plan to liftoff
- k Abort and teleop to liftoff
- l Create next grasp fixture
- m Drop piece

(c) Annotated log of piece 2.

Fig. 6: Anatomy of the Debris task.

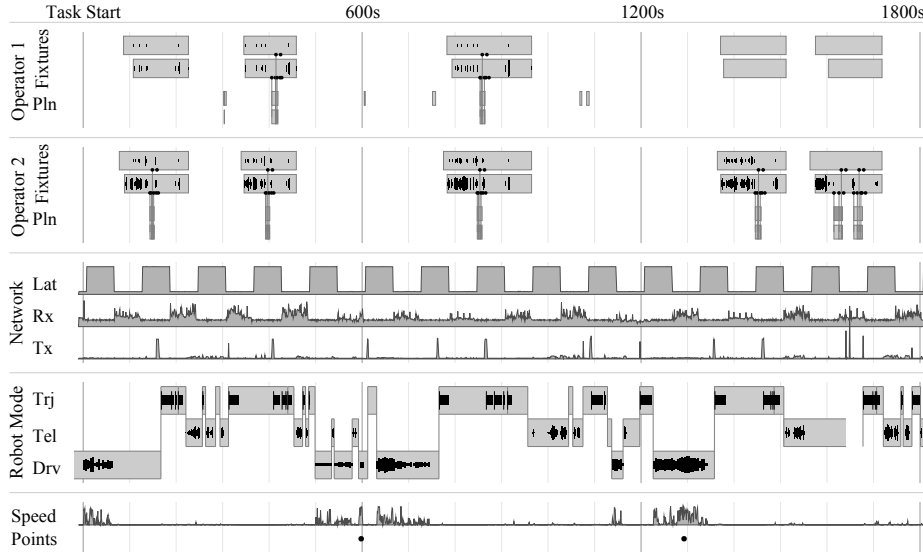


Fig. 7: Timeline of Door task. Each opening required two fixtures (a *GraspCylinder* for the handle, and an *Axis* for turning constraint); two operators cooperated to adjust fixtures and request plans. Two events resulted in premature door closings, necessitating repeated opening attempts: at 325s, a strong wind gust blew the first closed, and at 1547s, an errant teleoperation command caused the third to slip from CHIMP’s control.

3.2 Door Task

Door Setup. The task required sequentially opening and traversing three doors: the first a “push” door (1 pt.), the second a “pull” door (1 pt.), and the third a “pull” door with spring closure (1 pt.). All doors had identical lever-style handles.

Door Approach. We used the *GraspCylinder* and *Axis* fixtures to approach, grasp, and turn each door handle. See Fig 3 for examples of the axis fixture, which constrained the planner to move the valve handle body about a fixed axis. During execution, this trajectory segment was executed with a Cartesian force controller as described in Sec. 2. Subsequent manipulation of the doors (pulling and pushing) was performed via a combination of (a) pre-selected arm configurations and (b) gripper Cartesian-space teleoperation. Traversing the third door required positioning CHIMP in such a way that the door was actively held open while it was being traversed.

Door Results. See Figure 7 for a time breakdown of the door task. We achieved two points in the allotted 1800s. CHIMP successfully actuated the door handles in all 5 attempts, but suffered two events which lead to premature door closures on the first and third doors, requiring extra time.

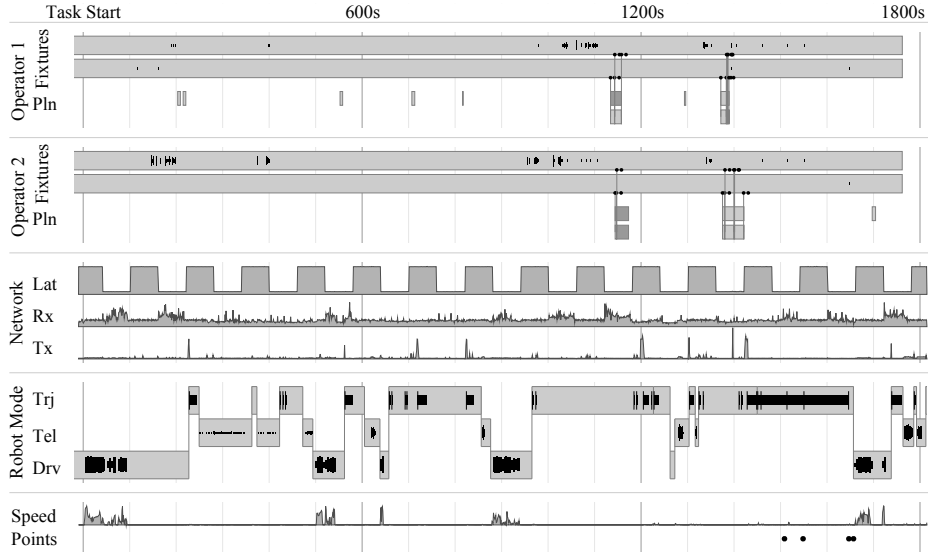


Fig. 8: Timeline of Wall task. The drill was successfully grasped and test-actuated around 550s. The second operator made continual adjustments to the polygonal-path fixture, especially between 950s-1050s before the first plans. During supervision of the first execution at 1200s, the operators visually detected that the cut was not as deep as desired, so the fixtures were re-adjusted. Cutting proceeded from 1425s-1650s.

3.3 Wall Task

Wall Setup. The task required grasping a cordless cutting instrument and using it to cut a prescribed triangle shape in drywall. Teams could choose between a drill loaded with a side-cutting bit, or a small circular reciprocating saw. Each of the three edges of the triangle successfully cut (without damaging the wall outside the lines) was worth one point.

Wall Approach. We used the `GraspCylinder` fixture to approach the drill, along with precision nudges to precisely grasp it so that the trigger was reachable. Actuation of the tool was visually inspected using a trigger-actuated light. We used the `Box` and `RotaryCutTool` fixtures to model the volume and cutting bit of the drill, and used the `PlanarPath` and `Vector` fixtures to fully specify the location of the triangle shape on the wall, along with the approach direction and distance. Constrained planning was used to compute a full trajectory to perform all steps of the cut.

Wall Results. See Figure 8 for a time breakdown of the Wall task. We achieved four points in 1647s. After the first attempt, during supervision, the operator team determined that the bit may not have sufficiently punctured the wall; the puncture distance was adjusted, and after replanning, the trajectory was allowed to run to completion. Once the drill was grasped and the path annotated, the operators assumed only supervisory roles.

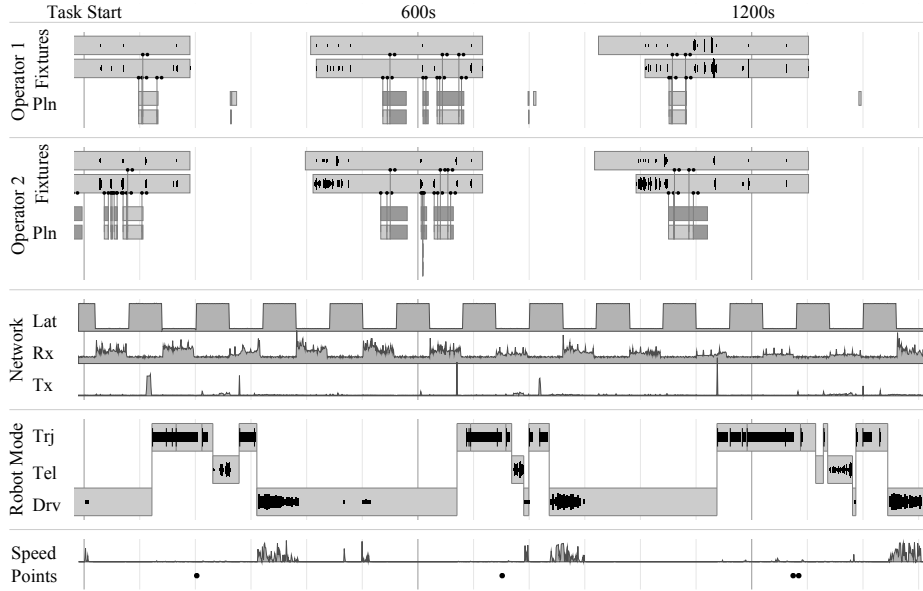


Fig. 9: Timeline of Valve task. Each valve was annotated using two fixtures. After the resulting motion was previewed, the trajectory was sent to the robot (at 110s, 670s, and 1140s) for supervised execution. After each successful turn, the gripper was extracted using teleoperation (e.g. at 340s) and the arm was reset to a driving posture (e.g. at 385s). The robot was then driven to the next valve, and the process repeated.

3.4 Valve Task

Valve Setup. The task required grasping and turning three valves (a 90-degree lever valve, a large circular valve, and a small circular valve). See Figure 2. Each of the three valves completely turned (360 degrees for the circular valves) earned one point.

Valve Approach. We used the `GraspCylinder` fixture to describe the grasp strategy for each valve handle body. We then used the `Axis` fixture to label the axis of rotation of each valve. See Fig. 3 for examples of these fixtures for the valve task. Constrained planning was used to compute a full trajectory to turn each valve. During execution, this trajectory segment was executed with a Cartesian force controller as described in Sec. 2.

Valve Results. See Figure 9 for a time breakdown of the Valve task. We achieved four points in 1275s.

3.5 Hose Task

Hose Setup. The task required retrieving a hose from a wall-mounted spool, and transferring it for several meters (1 pt.), touching the hose nozzle to a wye (1 pt.) and threading it onto the wye (1 pt.).

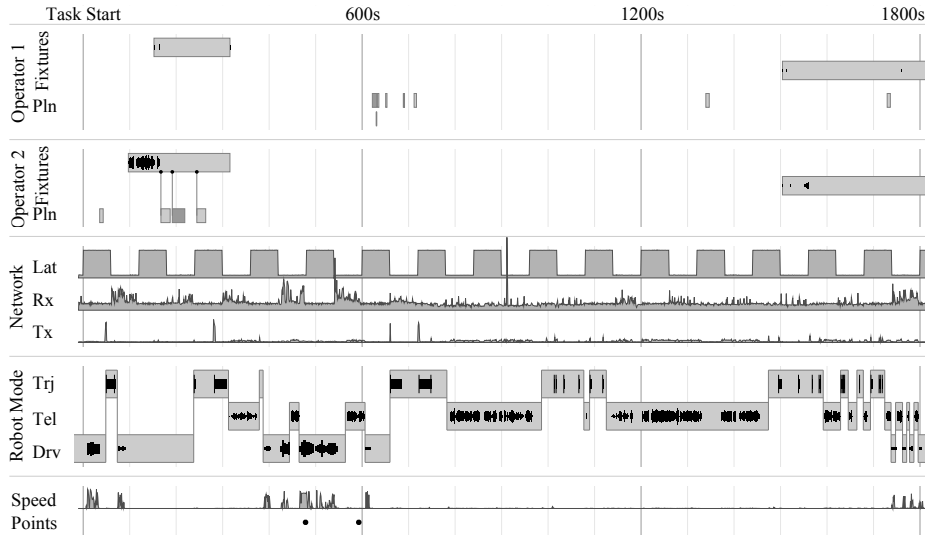


Fig. 10: Timeline of Hose task. The hose was grasped using a grasp strategy fixture at 380s, and the first two points were achieved by 593. The remaining time was spent attempting to thread it onto the wye, with no success.

Hose Approach. We used the generic grasp fixture to retrieve the hose, and quickly transferred and touched the nozzle to the wye. In limited testing, we hadn't found a robust way to accomplish the threading component.

Hose Results. See Figure 10 for a time breakdown of the Hose task. We achieved two points in the first 593s, and spent the remainder of the task time attempting to complete the third subtask.

4 Main Experimental Insights

Team Tartan Rescue placed third in the competition, and achieved 16 out of a possible 20 points on the five manipulation tasks (see Figure 11). We were also the only team in the competition which was not penalized for an intervention.

The time spent for each task is shown in Figure 12. Note that this allocation does not account for pipelining; when multiple operators were performing different actions simultaneously, the one deemed in the critical path was counted.

Our approach forms a strong foundation for human-guided manipulation applicable to disaster response scenarios.

Virtual Fixtures. We found the development and workflow of virtual fixtures to be at the appropriate level of generality and extensibility for the problem we addressed at the DRC Trials competition. As first-class spatial objects, operators found it straightforward to reason about their representation during the tasks.

Operator Experience. In contrast to a fully-autonomous system, we found performance was correlated with operator training. Over time, the operators

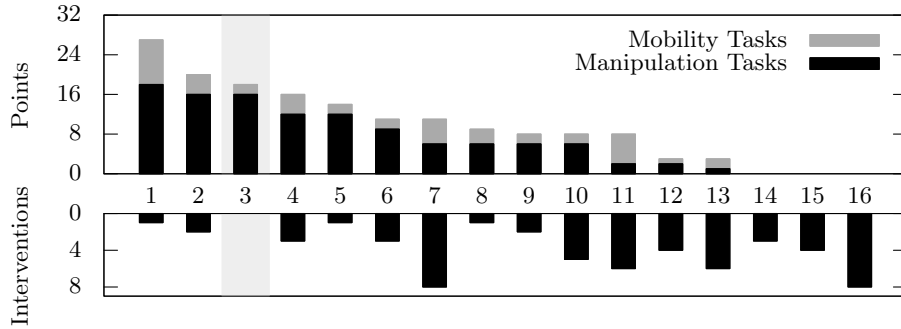


Fig. 11: Scores of each team at the DRC Trials, ranked by total points achieved. Team Tartan Rescue placed third with 18 points (tied for second with 16 points on the five manipulation tasks), and was the only team with zero interventions.

	Comms	Driving	Fixturing	Planning	Traj Exec	Teleop
Debris	7.6%	15.6%	16.5%	11.6%	30.1%	18.5%
Door	11.1%	24.5%	15.1%	7.5%	23.0%	18.8%
Wall	2.4%	16.6%	17.4%	8.2%	38.3%	17.0%
Valve	10.2%	16.9%	15.2%	23.7%	29.6%	4.4%
Hose	3.0%	13.2%	5.3%	9.7%	25.9%	42.9%

Fig. 12: Time breakdown of each task.

learned heuristics for task parameters (e.g. base placements, grasp orientations, etc.) that lead to fast and robust solutions.

Multi-Step Planning Robustness. When solved naïvely, multi-step plans can often fail by committing to choices early that preclude efficient solutions to later steps. This happened occasionally during the trials (e.g. during debris piece 2 from Figure 6(c)). A solution to this problem may improve planning success rates, and is a promising area for future work.

Pipelining. We exploited pipelining between locomotion, fixturing, planning, and trajectory execution to improve our task completion times.

Failsafes. During trials, failed or long-running plans or executions were superseded by end-effector teleoperation or joint-level control. This strategy allowed for increased robustness and execution speed.

4.1 Future Directions

Our current approach and implementation is a first step towards developing a framework for guided manipulation. We are excited about two directions of future work: *autonomy* and *expressiveness*.

Towards greater autonomy. Our current framework relies completely on the operator for the deployment of virtual fixtues. By relying on the operator’s

spatial awareness, we are able to execute complex manipulation tasks with little perception: our system uses unstructured voxel worlds for collision avoidance and does not currently perform semantic perception.

Our framework does expose the scaffolding for semantic perception and learning. Given semantic information from a perception system, like objects, handles, and door kinematics, the system can automatically initialize fixtures that are cached or learned from previous demonstration.

Towards greater expressiveness. The expressiveness of virtual fixtures depends synergistically both on the capabilities and preferences of the operator, and on the capabilities of the underlying planning algorithms. Currently, we are restricted to virtual fixtures expressed as Cartesian task-space regions, and planning requests as fixed sequences of fixtures. In the future, we envision adding branching, giving the planner options to choose from, as well as more complex constraints related to stability and sensor visibility.

References

1. J. Bagnell, F. Cavalcanti, L. Cui, T. Galluzzo, M. Hebert, M. Kazemi, M. Klingensmith, J. Libby, T. Y. Liu, N. Pollard, M. Pivtoraiko, J.-S. Valois, and R. Zhu. An integrated system for autonomous robotics manipulation. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2955–2962, Oct. 2012.
2. D. Berenson, S. Srinivasa, D. Ferguson, and J. Kuffner. Manipulation planning on constraint manifolds. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2009.
3. D. Berenson, S. Srinivasa, and J. Kuffner. Task space regions: A framework for pose-constrained manipulation planning. *The International Journal of Robotics Research*, 30(12):1435–1460, 2011.
4. A. Bettini, P. Marayong, S. Lang, A. Okamura, and G. Hager. Vision-assisted control for manipulation using virtual fixtures. *Robotics, IEEE Transactions on*, 20(6):953–966, Dec. 2004.
5. T. Debus, J. Stoll, R. D. Howe, and P. Dupont. Cooperative human and machine perception in teleoperated assembly. In D. Rus and S. Singh, editors, *Experimental Robotics VII*, volume 271 of *Lecture Notes in Control and Information Sciences*, pages 51–60. Springer Berlin Heidelberg, 2001.
6. R. Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Robotics Institute, Aug. 2010.
7. R. C. Goertz. Manipulators used for handling radioactive materials. 1963.
8. A. Kron, G. Schmidt, B. Petzold, M. Zah, P. Hinterseer, and E. Steinbach. Disposal of explosive ordnances by use of a bimanual haptic telepresence system. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 2, pages 1968–1973 Vol.2, Apr. 2004.
9. A. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow. In *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, pages 1–8, Mar. 2012.
10. S. Park, R. D. Howe, and D. F. Torchiana. Virtual fixtures for robotic cardiac surgery. In *Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI '01*, pages 1419–1420, London, UK, UK, 2001. Springer-Verlag.

11. M. Quigley, B. Gerkey, C. K., J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *Proc. Open-Source Software workshop, IEEE International Conference on Robotics and Automation, 2009. (ICRA)*, 2009.
12. G. Rodriguez and C. Weisbin. A new method to evaluate human-robot system performance. *Autonomous Robots*, 14(2-3):165–178, 2003.
13. L. Rosenberg. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, pages 76–82, Sept. 1993.
14. T. Stentz, H. Herman, A. Kelly, E. Meyhofer, G. C. Haynes, D. Stager, B. Zaccaro, J. A. Bagnell, J. Brindza, C. Dellin, M. George, J. Gonzalez-Mora, S. Hyde, M. Jones, M. Laverne, M. Likhachev, L. Lister, M. Powers, O. Ramos, J. Ray, D. Rice, J. Scheifflee, R. Sidki, S. Srinivasa, K. Strabala, J.-P. Tardif, J.-S. Valois, J. M. Vande Weghe, M. Wagner, and C. Wellington. CHIMP, the CMU highly intelligent mobile platform. *Submitted*, 2014.
15. R. Taylor. Robots as surgical assistants: Where we are, wither we are tending, and how to get there. In E. Keravnou, C. Garbay, R. Baud, and J. Wyatt, editors, *Artificial Intelligence in Medicine*, volume 1211 of *Lecture Notes in Computer Science*, pages 1–11. Springer Berlin Heidelberg, 1997.
16. J. Trevelyan, S.-C. Kang, and W. Hamel. Robotics in hazardous applications. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 1101–1126. Springer Berlin Heidelberg, 2008.
17. T. Xia, S. Leonard, I. Kandaswamy, A. Blank, L. Whitcomb, and P. Kazanzides. Model-based telerobotic control with virtual fixtures for satellite servicing tasks. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1479–1484, May 2013.