Shared Autonomy: An Operator's Perspective

Joshua James
TRACLabs
Houston, TX, USA
Email: jjames@traclabs.com

***Abstract*—This paper analyzes the human-robot boundaries chosen by team TRACLabs at the DRC Trials from the perspective of the human operator and presents ideas for future improvements based on our experience at the Trials.**

I. Introduction

The DARPA Robotics Challenge (DRC) [1] is designed to promote the development of technologies for semi-autonomous robots capable of service in disaster scenarios. The DRC is composed of three parts: the Virtual Robotics Challenge (VRC), DRC Trials, and DRC Finals. TRACLabs started as a track B team (software-only, funded) during the VRC, and has performed well enough to earn a physical robot [2] and compete to the Finals [3], scheduled for June 2015.

Atlas robots, developed by Boston Dynamics (BDI), were provided to several teams for use in the DRC. Atlas has 28 hydraulic actuators: six per leg, six per arm, three in the back, and one in the neck. Every joint has torque and position sensing. An inertial measurement unit (IMU) mounted to its pelvis provides robot pose information. The Multisense-SL head, developed by Carnegie Robotics, provides most of the non-proprioceptive sensing. The head contains a 180° LIDAR on a continuously rotating spindle and a pair of color cameras each with an 80° FOV. Because the head cannot turn laterally, two side-facing "situational awareness" (SA) cameras were added. The SA cameras have highly warped fish-eye lenses with a 185° FOV to cover most of Atlas' surroundings. In addition to hardware, BDI also provided locomotion and balancing behaviors for use on the Atlas robots.

Team TRACLabs is fairly small, with roughly four primary developers. Thus, our approach to the DRC Trials was to reduce development time by relying on the human operator for high-level decisions, introducing autonomy on the robot only where necessary. To that end, we chose to develop low-level [4] autonomous behaviors (essentially tools to aid the operator) rather than attempting to create high-level autonomy capable of solving the competition completely. The following sections analyze our placement of the robot-human boundary in three areas relevant to the competition (perception, planning, and control), in order of use by the operator.

II. Perception

When the operator is responsible for most analysis and decision making, access to good perception data is one of the most important aspects of the system. People typically out-perform robots at interpreting noisy or heavily discretized image data, when presented in human-intelligible form. Therefore, the perception automation we used at the Trials consisted mostly of basic camera image processing and spatial aggregation of laser scans. The resulting point clouds and images were discretized, compressed, and relayed to the operator for analysis. We found that some of the sensors were more helpful to high-level human autonomy than others.

The point cloud from the LIDAR was useful for both step and manipulation planning, but did present some issues. Scans are planar, and so require aggregation over time to yield a coherent 3D world representation. Due to the constant aggregation, unmodeled changes in the world (*e.g.*, an object in the environment moves or is moved) cause blurring in the point cloud, making those areas difficult to interpret. This effect is magnified by the LIDAR's rotation around a fixed axis, which results in dense readings near the center that get sparser further away. The sensed region is large, but interpretation is more difficult in the "laser peripheral vision". These issues often made it necessary to obtain a fresh point cloud, particularly before a manipulation, and freeze it to prevent contamination.

The front-facing head cameras were useful for general awareness, especially for detecting unsafe contact forces (*e.g.*, pushing on a wall hard enough to shift the torso makes the camera images visibly shift). However, they were nearly useless for manipulation, with little overlap between their field of vision and the kinematic workspace of the robot's arms.

The angled SA cameras proved to be the most useful sensor for correcting manipulation errors caused by joint sensing errors and poor control, and were used heavily during the competition. Despite being fish-eyed and difficult to interpret, they were still the best choice, as the other cameras were often unable to see the objects being manipulated.

Not being able to look at and manipulate objects at the same time is a central issue with Atlas that greatly impairs an operator's ability to plan well. The most significant addition that could be made to Atlas to improve shared autonomy is the ability to directly gather perception data anywhere in the robot's configuration space. The addition of a neck yaw joint (allowing Atlas to look side-to-side without turning its back) would solve most of these problems, though extra sensors on the legs and arms for additional data on stepping and manipulation targets respectively would be helpful as well.

Beyond hardware modification, more perception autonomy could be shifted to the robot side to aid the operator. Given a sufficiently large database of 3D models, many objects could be automatically recognized and highlighted for the operator to speed up the manipulation process. Since the robot has access to the dense point cloud, it may be able to find objects invisible to the operator due to discretization and compression.

III. Planning

To plan effectively, the operator must have access to a good set of tools. At the Trials, our operator primarily used low-autonomy setpoint-based actions. The operator provided a setpoint to be reached at a given time, and the robot decided how to make it happen. The four tools we used most often were walking, single-stepping, Cartesian-space manipulation, and joint scripting.

The walking tool provides the ability to move the base of the robot to a specified 2D pose. It employs a simple flat-ground footstep planner to generate steps for the stepping controller provided by BDI. This is useful for autonomously traversing flat portions of the environment, as well as rough alignment of the robot to manipulation targets.

The single-stepping tool allows the operator to specify a single target step as a 6-DoF pose. This allows for more precise movement of the robot and is used both for aligning the kinematic workspace of Atlas' upper body with objects to be manipulated and crossing rough terrain.

The Cartesian-space manipulation tool moves end-effectors to desired target poses using a subset of the joints in Atlas' upper body, either following a straight line or an arc in Cartesian space. This was the tool we used for all of the manipulation we did during the Trials.

The joint scripting tool enables the operator to execute joint-space scripts created at run-time, as well as predefined scripts, to make planning faster and more consistent. Run-time scripting was used in development, but not during the competition. Some examples are an arm configuration that is a good starting point for manipulation and a configuration in which it is safe to turn off the robot.

Overall, this setup worked well but was fairly slow during manipulation. Direct teleoperation of the end-effectors by a human would have been significantly faster, but was not viable during the Trials due to networking restrictions. Adding higher-level planning to the robot side would allow us to operate more efficiently under poor networking conditions, but may still not be as fast or reliable as direct teleoperation. To perform well under all bandwidth conditions, it is therefore best to use a sliding autonomy system. Such a system would allow the command hierarchy to be accessed at all levels (*e.g.*, "solve the task" vs. "turn that valve" vs. "move these joints here") and let the operator choose the appropriate level of autonomy based on the current situation [5].

IV. Control

All plans, whether they are created by a robot or a human, are contingent on the ability to execute them successfully. At the Trials, we primarily executed plans with low-level joint position control performed by the robot, augmented with sporadic "goal-space" human operator control.

The joint position control we used at the time of the competition was very poor. Because we were using simple PD controllers, there was often significant steady state error due to gravity. The encoder measurements also often did not accurately reflect the actual joint positions, further impairing control. The combination of these effects created a situation in which the robot never reached its setpoints, and we often didn't know where the end effector actually was with respect to the world, unless we were able to see it in the point cloud data or a camera.

The goal-space feedback control performed by the human operator allowed us to overcome these low-level control issues. Humans intuitively think of error in terms of high level goals, which is why teleoperation is often more successful than robot-planned solutions. For example, when turning a valve manually, you are primarily concerned with how well the valve is turning, rather than how far off your joints are from where you thought they should be if you closed your eyes and imagined turning the valve (*i.e.*, the typical robotics approach). In this case, the operator monitored the true error based on perception and provided correctional feedback in the form of new commands if the errors proved detrimental to the progress of the task. Even slowed down by bandwidth restrictions, this high-level feedback significantly ameliorated the low-level control issues.

Though we have greatly improved our joint sensing and position control since the Trials, we still frequently find the operator's intuitive goal-based error correction to be useful, and direct teleoperation remains the fastest option for task completion. Moving goal-based control to the robot side would greatly enhance robot autonomy, but is beyond the current state of the art.

V. Conclusion

Despite being a small team, we succeeded at the Trials by developing low-level tools instead of high-level autonomy. In general, we used the robot for things that can be modeled perfectly, can be executed reliably, and must be done in real time. We generally used the human operator for planning, adapting to new situations, data analysis, and using intuition to make corrections in goal-space. Good sensor placement can greatly improve human perception and planning ability, and should be taken into account when designing future robots. To maximize flexibility with respect to changing bandwidth conditions, a complete but hierarchical set of tools at various autonomy levels should be provided that are accessible by the human operator at any level. To that end, future development will include object recognition and automated completion of larger task components, ideally supported by robot-side goal-space feedback control.

References

[1] DARPA. (2012, Apr) Darpa robotics challenge. [Online]. Available: http://www.darpa.mil/Our_Work/TTO/Programs/DARPA_Robotics_Challenge.aspx

[2] ——. (2013, Jun) Members of top nine software teams move forward from darpas virtual robotics challenge. [Online]. Available: http://www.darpa.mil/NewsEvents/Releases/2013/06/27.aspx

[3] ——. (2013, Jun) The darpa robotics challenge continues june 2014 in southern california. [Online]. Available: http://www.darpa.mil/NewsEvents/Releases/2014/06/26.aspx

[4] R. Parasuraman, T. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 30, no. 3, pp. 286–297, May 2000.

[5] S. Lenser and C. Jones, "Practical problems in sliding scale autonomy: a case study," in *Unmanned Systems Technology X*. iRobot Corp, 2008.