Autonomous Mission with a Mobile Manipulator - A Solution to the MBZIRC

Autonomous Mission with a Mobile Manipulator - A Solution to the MBZIRC

Jan Carius, Martin Wermelinger, Balasubramanian Rajasekaran, Kai Holtmann, and Marco Hutter

Abstract This work presents the system and approach we employed to tackle the second challenge of the Mohamed Bin Zayed International Robotics Challenge (MBZIRC)[1]. The goal of this challenge is to find a tool panel on a field, pick an appropriate wrench from the panel, and operate a valve stem therewith. For this purpose we use a task-oriented field robot, based on Clearpath Husky with a customized series elastic arm, that can be deployed for versatile purposes. However, to be competitive in a robotic challenge, further specialization and improvements are necessary to achieve a certain task faster and more reliably. A high emphasis is put on designing a system that can operate fully autonomously and independently respond if a subtask was not executed successfully. Moreover, the operator can easily monitor the system through a graphical user interface and, if desired, interact with the robot. We present our algorithms to explore the field, detect the panel, and navigate to it. Furthermore, we use a support vector machine based object detection method to locate the valve stem and wrenches on the panel for visual servoing. Finally, we show the advantages of a force controllable manipulator to handle the valve stem with a tool. This system demonstrated its applicability when fulfilling the entire task fully autonomously during both trials of the Grand Challenge of the MBZIRC 2017.

1 Introduction

Robotic competitions and challenges give a strong motivation to foster innovation and research and to aim for a long-term goal. They support development and real-

Jan Carius
Robotic Systems Lab, ETH Zurich, e-mail: jan.carius@mavt.ethz.ch

Martin Wermelinger
Robotic Systems Lab, ETH Zurich, e-mail: martin.wermelinger@mavt.ethz.ch

[1] See http://www.mbzirc.com/challenge

world verification of single components, but also force to integrate those components into a complete system. This may be a reason why these contests gained more and more popularity over the past years and spread over various areas of robotics. Contests like *Cybathlon* [16], *DARPA Robotic Challenge (DRC)* [8], *ELROB* and *euRathlon* [17], *European Robotic Challenges (EuRoCs)* [18], *RoboCup* [12], and *RoCKIn* [13], to mention only a few, show the sate of the art but also catalyze new developments. Furthermore they present a great opportunity to promote the progress in robotics to a broad audience. However, robotic challenges may also drive the technological innovation into an "reductionist design" [2], where simplistic but custom-tailored solutions are applied instead of generic approaches. Finally they expose clear weaknesses of current systems that require help from several humans to be set up and supervised.

This work presents the approach and system design that we used to master the second challenge of the MBZIRC 2017. The task of this challenge is to use an unmanned ground vehicle (UGV) to locate a wrench panel on a 60 x 60 m outdoor field, navigate to it, select and grip a suitable wrench hanging on the panel, and turn a valve stem 360 ° therewith. A special focus of this challenge lies on complete autonomous behavior, since any intervention by a human operator will directly result in significant score penalties. We decided to utilize a generic mobile manipulation platform that can be used for a variety of tasks. The modifications made on hardware and software are modular and complement the base system.

The remainder of this work will cover our contributions to accomplish the challenge task: design of a mobile robot for handling wrenches (Sec. 2); autonomy and mission state machine (Sec. 3); detection of the panel (Sec. 4); exploration, navigation, and positioning (Sec. 5); object detection and visual servoing (Sec. 6); valve manipulation (Sec. 7); and finally we conclude with the performance during the Grand Challenge.

2 Hardware

The system used for this robotics challenge consists of the four-wheeled skid-steer platform Husky (Clearpath Robotics), equipped with a custom six degree of freedom (DoF) manipulator arm (see Fig. 1 left). For state estimation, localization, and navigation in unknown environments the UGV is equipped with wheel encoders, a Microstrain 3DM-GX4-15 inertial measurement unit (IMU), a Velodyne HDL-32E light detection and ranging (LIDAR) sensor, and optionally a Garmin GPS-18x global positioning system (GPS) receiver. The arm is an advancement of the previously presented ANYpulator [1] with newly three DoF allocated at the wrist of the manipulator. All joints of the arm are actuated by the same high-performance series elastic actuator (SEA) units. These actuators, called ANYdrives [10], are composed of a brushless high-torque motor, harmonic drive gear, and a torsional spring. The usage of SEAs adds inherent compliance to the system, making it especially suitable for interaction tasks such as valve manipulation. The computational payload for

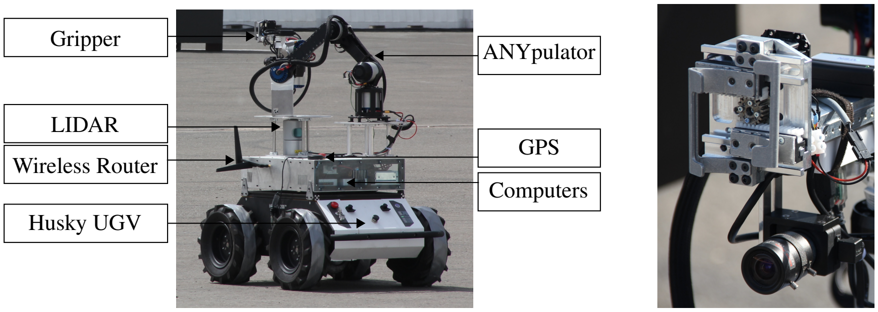


Fig. 1 Left: Integrated mobile manipulator platform with six DoF arm. **Right:** Clamp design for grasping wrenches with attached camera for visual servoing.

navigation, arm control, and perception is distributed on three on-board computers, which connect to an operator computer through a wireless network.

Specifically designed for the task of the challenge is the end-effector tool for wrench grasping (see Fig. 1 right). The gripper's mechanical design is optimized to provide a lightweight but effective tool for holding the shafts of standard wrenches. A hollow aluminum structure attaches to the final joint of the robot arm and provides mounting points for a camera and a clamping mechanism. A single Dynamixel MX-64 Motor drives two sheet metal clamps through a rack and pinion mechanism. This ensures that the closing position is centered laterally with the end-effector reference frame. Additionally, the clamps are designed to center the head of a gripped wrench with the rotation axis of the last joint. This facilitates pure rotation motion of the tool, e.g. for turning a screw or the valve stem. Sensing of motor current and position allows to determine whether or not an object is successfully held by the gripper. For visual servoing, a calibrated monocular camera (PointGrey Chameleon 3) is attached to the aluminum shaft below the clamps, such that a grasped wrench does not obstruct the view of the camera.

3 Autonomy and Mission Control

A typical mission that we expect our robot to fulfill involves several sequential steps such as exploration, detection of the manipulation site, navigation, positioning, and manipulation itself. Tasks of this complexity call for a modular mission design since each sub-task may fail and needs to be repeated. Fig. 2 displays our mission state flow diagram for the second task of the MBZIRC. Each state is responsible for a specific sub-task and is accompanied by a supervision module (not shown in the figure) that decides if the completion was successful or not. In case of failure specific recovery behaviors may be executed or the state machine simply transitions back to the previous task and starts another attempt. This architecture also allows the

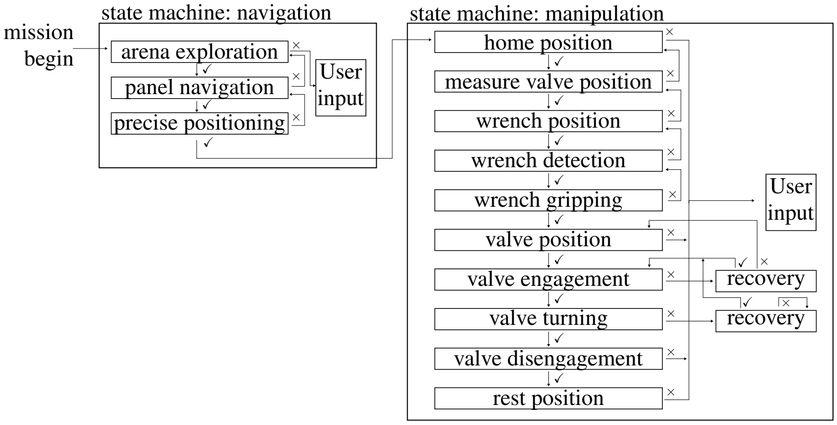


Fig. 2 State flow during the MBZIRC mission. Each state may succeed (✓) or fail (×). On failure, the robot will execute the previous task again or request user input.

interruption of a current task on predefined conditions, for example when the battery runs low or the operator requests the mission to stop.

The operator can monitor the robot through an additional operator computer, which is connected to the robot's wireless network. A graphical user interface (GUI) displays information from all relevant software modules and diagnostic information such as battery charge state and temperatures. Additionally, a visualization of the constructed environment map and the image stream from the camera are displayed to the operator. At any time, the operator may interrupt the autonomous mission and provide manual commands through a joystick or through the GUI. This empowers an user to issue specific commands such as closing or opening the gripper, switching to a specific mission state, or controlling the manipulator or mobile platform.

All mission critical software is running on the robot itself and is independent from the quality of the WiFi network. Therefore, even under limited network connectivity,

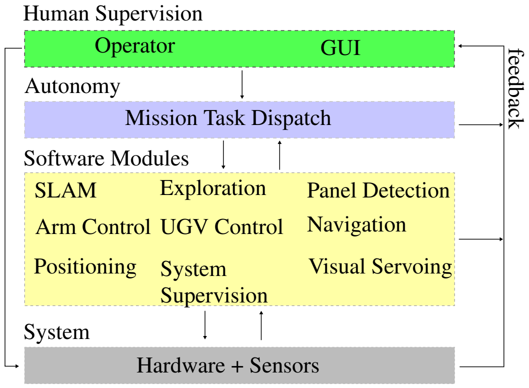


Fig. 3 Hierarchical software architecture and abstraction. The operator can interact with the mission task dispatcher or teleoperate the system directly.

the robot maintains its autonomous behavior. For safety, the robot pauses if there is a connection time-out to the joystick.

4 Wrench Panel Detection

An important element for succeeding in the navigation part of the MBZIRC is the correct identification of the wrench panel and its subsequent pose estimation. Our method relies on 3D point clouds acquired from a laser scanner.

4.1 Point Cloud Processing

The panel detection software operates on any kind of point cloud (e.g., constructed map or raw scan) with the only requirement that the vertical direction of the cloud is known. For the MBZIRC, our setup directly uses raw scans, obtained by assembling laser data from one revolution of the scanner.

To identify the panel, we make use of the known geometric dimensions of the rectangular sides as given by the challenge description. Our algorithm can identify any planar side of the panel by executing the following operations:

1. The point cloud is cropped to a box with side length of 10 m around the robot. This reduces the computational requirements and prevents detection of false positives.
2. A normal vector is assigned to each point through principal component analysis of the covariance matrix from neighbors of the query point.
3. Region growing using conditional Euclidean clustering assigns each point to a cluster, thereby segmenting the point cloud. Two points share the same cluster if they are sufficiently close (Euclidean distance) and their normals are aligned.
4. In each cluster, random sample consensus (RANSAC) [6] plane extraction singles out individual planes. We reject non-vertical planes and those that lie outside of our search space (specified as a polygon corresponding to the arena geometry).
5. For each of the remaining planes, we check if its dimension matches any of the rectangular sides of the panel. Finally, we confirm that the point distribution across the plane is approximately uniform to reject any remaining artifacts.

4.2 Pose Estimation

If all of the above tests are successful, the identified plane, together with its orientation and position, is passed to an estimation module which keeps track of several detections. The individual detections (*measurements*) of the panel pose are fused to potential matches (*candidates*). Each candidate *i* holds an internal state, containing the pose of the front plane of the panel with respect to a map-fixed reference frame and an improper probability $p_i \in [0,1]$ indicating how likely it is the true match. New measurements will update existing candidates if there exists a sufficiently close one, otherwise a new candidate will be initialized with probability p_{init}.

(a) **Initializing Candidates:** Upon construction of a new candidate, the pose information of the measurement is transformed to an equivalent pose of the panel's front side by applying a suitable translation and rotation. If the initial measurement is ambiguous, i.e., the algorithm detected the side of the panel, one cannot decisively infer if the detection corresponds to the right or left part. In this case, an arbitrary choice is made and the state of this candidate is set to ambiguous.

(b) **Updating Candidates:** The first step in updating a candidate is again transforming the measurement to an equivalent pose of the front side of the panel. In ambiguous cases, the rotation is chosen such that it supports the current hypothesis. If a previously ambiguous candidate receives an update by a conclusive measurement (e.g., the front or back side), the ambiguity can be resolved. This potentially involves flipping front and back side if the wrong initial choice was made. The position is updated through a first order filter and the rotation through spherical linear interpolation and the probability is increased by p_{update}.

At each iteration, the candidate manager reorders its list in descending probability order, normalizes the probabilities such that the highest does not exceed 1.0, and removes candidates with probabilities below a given threshold. We accept the most likely candidate 0 if its probability $p_0 > p_{\text{threshold}}$ and it is significantly more likely than the next best candidate $p_0 - p_1 > p_{\text{diff}}$, if any. The parameters fulfill

$$0 < p_{\text{init}} + p_{\text{update}} < p_{\text{threshold}} < p_{\text{init}} + 2p_{\text{update}} < 1 \ , \tag{1}$$

hence at least three detections are necessary for acceptance of a candidate.

4.3 Evaluation

Our approach correctly identifies the pose of the wrench panel from a distance of 10 m, which is approximately the point at which the rectangular sides become visible in the point cloud visualization for a human observer.

The computational load of processing point cloud data allows update rates of 0.6 Hz on an Intel i7 processor, which places an upper bound on the speed at which the robot can drive past the panel without missing it. In practice, the maximum speed of our ground vehicle (1.0 m/s) does not cause any problems in this respect.

A major difficulty for the detection algorithm arises from occlusions in the laser scan. Our scanner is protected by a cover which is held by three vertical pillars, casting conic shadows with opening angles of 7°. If the panel happens to fall inside this region, the occlusions render the detection difficult or at times impossible.

Apart from detections in the laser scan, the estimation module accepts panel measurements from several sources. While we did not employ this in the actual challenge, it would have been possible to take input from a user or measurements made by drones about the potential position of the panel. The rejection mechanism of single false positives proves essential in many cluttered environments.

5 Navigation

We applied a custom navigation routine to drive and position the robot in front of the wrench panel. This routine makes use of underlying iterative closest point (ICP) based localization and mapping [14, 15] and the ROS navigation stack[2] for path planning and following. The navigation is divided into two main phases, first locating the panel in an exploration phase and secondly positioning the robot relative to it.

5.1 Exploration

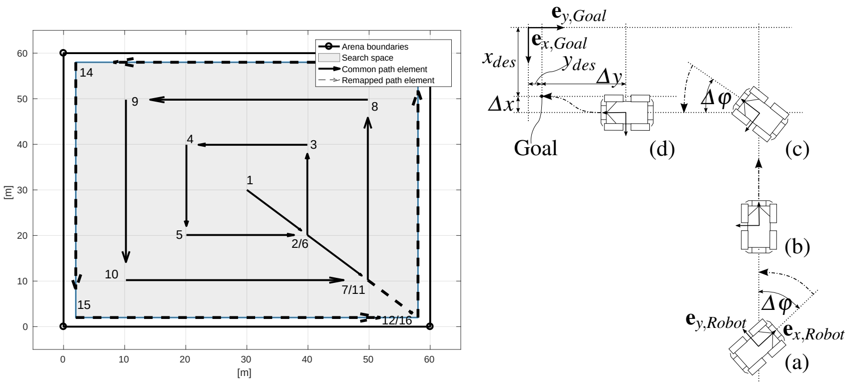


Fig. 4 Left: Waypoints for complete coverage of rectilinear search space. **Right:** The positioning procedure: (a) Facing, (b) Approaching, (c) Turning, and (d) Centering.

The panel has to be located before the robot can position itself precisely. Therefore, the exploration module creates a complete coverage path plan in a predefined rectilinear search space which is defined by the user via the input of GPS corner coordinates. This coverage path starts at the center of the search space and works its way towards the boundaries in counter-clockwise cycles around the middle point. The step size between those cycles thereby correlates directly with the detection range of the panel detection algorithm. An illustration of the coverage path plan is given in Fig.

[2] See http://wiki.ros.org/navigation

4 (left). Furthermore, this module comes with a set of recovery behaviors to increase overall robustness. They guarantee that exploration continues when encountering scenarios such as unreachable waypoints due to obstacles. For increased efficiency of the exploration, heuristic priority waypoints can be specified. They are tracked prior to the retracing of the inherent coverage path.

The exploration phase is usually the first stage in a mission and terminates once the concurrently running panel detection algorithm (Sec. 4) succeeds in identifying the manipulation site. In a subsequent transition phase, the robot navigates to a fixed position relative to the wrench panel. Once this intermediate goal is reached, the transition completes and the positioning phase is triggered.

5.2 Positioning in Front of the Wrench Panel

The purpose of this phase is to position the robot in a way that the panel is within the reach of the arm and subsequent manipulation tasks can be performed. To this end, a fixed sequence of maneuvers is executed. These maneuvers ensure that the desired goal pose of the robot in front of the wrench panel is reached reliably and accurately considering the skid-steer constraints of the mobile robot.

The positioning is divided into four stages: Facing, Approaching, Turning, and Centering. The schematic overview of these stages is illustrated in Fig. 4 (right). During the first three stages simple unidimensional velocity commands are applied, that is a constant rotation around the robot's z-axis in stage (a) and (c) and a linear velocity in stage (b). In the last stage both linear and angular velocities are controlled simultaneously. Thereby, potential residuals in lateral and longitudinal direction as well as rotational offsets from the previous stages are eliminated.

6 Object Detection and Visual Servoing

Manipulation of the valve using the appropriate wrench is the fundamental task of the mission. The key elements for succeeding in this task are correct detection of valve, precise estimation of depth, selection of the correct tool, and reliable visual servoing.

A support vector machine (SVM) [9] based object detection method using histogram of oriented gradients (HoG) features [4] is adopted in order to achieve robustness. HoG features are used as it captures the object's shape characteristic which is the most distinct property of the considered objects. A linear SVM model is trained using SVMLight [11] for each object i.e. valve, wrench, wrench box-end (referred to as wrench ring), and wrench open-end (referred to as wrench head). The training set contains images of the objects augmented for different illuminations to facilitate detection in different lighting conditions.

6.1 Valve Pose Estimation

We measure first the valve position and save it is as reference point for the subsequent manipulation, as the wrench panel layout is specified.

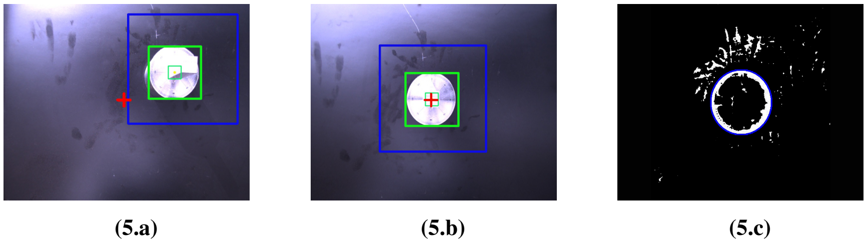


Fig. 5 Valve tracking (5.a) inside a region of interest (blue) to align the camera center (red) with the valve (green) (5.b). Detected valve contour (5.c) for depth estimation.

(a) **Valve Detection and Tracking:** The previously mentioned HoG based object detection method is used to detect the valve. The end-effector executes a predefined spiral motion pattern while searching for the valve. Once the valve is found with a certain confidence level, tracking starts using a Kalman Filter. Velocity commands are sent to the arm controller to reduce the alignment error between the center of the camera and the center of the valve as shown in Fig. 5.a. After aligning the camera with the valve (Fig. 5.b), the pose of the valve with respect to the robot's base can be computed using the current pose of the camera and the estimated distance between valve and camera.

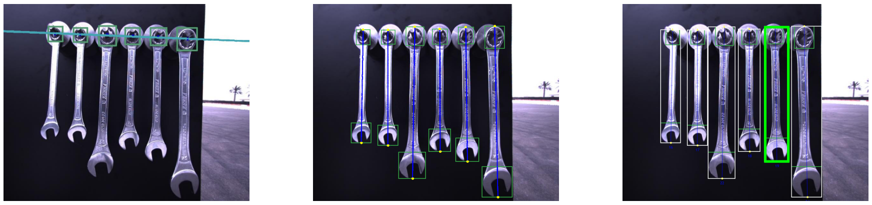
(b) **Depth Estimation:** The valve is segmented from the panel using adaptive thresholding [5] and its external contour is extracted using an algorithm based on connected components [19] as shown in Fig. 5.c. This gives us the size of the valve in pixels (d_{image}). Knowing the focal length (f_{cam}) of the camera, we determine the distance between the valve and camera (Z_{depth}) using the known size of the valve in metric standards (D_{world}) via the projective equation

$$Z_{\text{depth}} = \frac{D_{\text{world}} f_{\text{cam}}}{d_{\text{image}}} . \tag{2}$$

6.2 Wrench Selection

Corresponding to the given dimension of the valve stem, the appropriate wrench is selected according to its length with the following steps:

(a) **Wrenches View Camera Pose Estimation:** As the panel layout is specified, we can infer the camera pose where all wrenches are visible from the camera lens parameters (perspective view angle) and the estimated valve position.



(6.a) (6.b) (6.c)

Fig. 6 (6.a) Pin detection to estimate view orientation. (6.b) Wrench length estimation through head and ring detection. (6.c) Selected wrench (green) after length voting.

(b) **Orientation Alignment and Depth Estimation using Wrench Rings:** The wrench rings are detected using the mentioned SVM detector and the center points of the rings are accumulated from consecutive image frames. Once enough points are recorded, a line is fit over those points as shown in Fig. 6.a and its angle from the horizontal is calculated to correct the camera roll.

In order to be robust against positioning deviation of the UGV, we again estimate the depth from the wrench hanging pins. An approach similar to estimate the depth from the valve is used again. The distance (in pixels) between the consecutive pins in the image is equated to the distance (in metrics) between them from the panel layout using equation (2).

(c) **Wrenches Length Estimation and Selection:** The head and ring of every wrench is detected using the trained SVM models. The distance (in pixels) between the mid-points of the detected ends of the wrench is calculated as shown in Fig. 6.b and is converted to metric length.

In order to be robust against swinging of wrenches due to wind, a voting scheme is used. A wrench is selected in each image frame based on its estimated length. Each candidate is voted based on its frequency of occurrence as shown in Fig. 6.c.

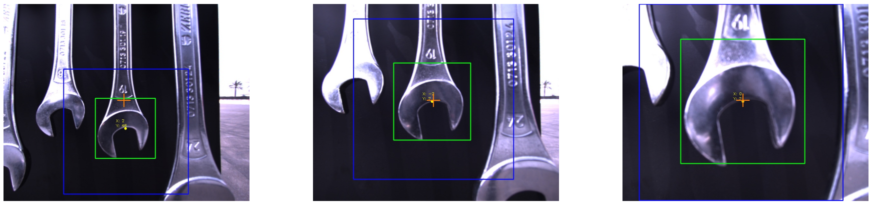
6.3 Wrench Grasping

Once the appropriate wrench is selected, it is approached by the gripper using visual servoing. Our method uses both position-based visual servoing (PBVS) and image-based visual servoing (IBVS) [3, 7] control techniques.

(a) **Approaching the Selected Wrench:** First, we use the estimated depth and a PBVS approach to move the gripper close to the selected wrench as shown in Fig. 7.a, from where it can be tracked.

(b) **Wrench Head Tracking:** IBVS is used to align the gripper with the wrench before grasping. During alignment, the gripper also approaches the wrench as shown in Fig. 7.b. A SVM model trained for the wrench head along with a Kalman Filter is used for detection and tracking.

While approaching, we calculate the ratio between the size of the wrench head and the size of the image frame. As we move closer this ratio increases and we can



(7.a) **(7.b)** **(7.c)**

Fig. 7 Wrench head tracking for grasping: The wrench head is detected (green) within an dynamic region of interest (blue) to align the camera (red) during approaching.

detect when the gripper is close enough to grasp the wrench as shown in Fig. 7.c. A first order low pass filter on the ratio is used to avoid false detection due to movement of the wrench in presence of wind.

(c) **Grasping:** Once the gripper has approached the wrench, the gripper closes slightly ensuring the wrench is bound within the claws. Then it moves down such that the wrench head is aligned to the rotation axis of the last actuator and does not occlude the camera view. The clamps close with a constant force and the final clamp position is used to detect if the grasping was successful.

7 Valve Manipulation

To fulfill the major part of the manipulation task it is sufficient to simply follow a desired end-effector trajectory. However, the ability to precisely measure and control the applied contact forces and torques demonstrates its usefulness during interaction of the gripper with objects. In particular, it shows its advantages in handling the valve stem with a wrench for both engagement and rotation.

7.1 Valve Engaging

Before engaging the wrench with the valve stem, we have to make sure that it is properly gripped. Though in the previous steps we ensure that the wrench is within the gripper, there is a chance that wrench slides down due to its weight. For this purpose we turn the gripper upside down, open the claws slightly to let the wrench head completely slide to the mechanical end stop, and close the gripper again. This ensures that the wrench head is at the correct position for engagement. Subsequently, the engaging procedure is executed.

First the gripper is placed in safe distance to the panel such that wrench head opening is pointing towards the valve stem edge, with an offset of 5 cm to the stem center. Next, we make contact between the wrench and the valve stem by first approaching towards the panel, afterwards to the center of the valve stem, and finally opening the gripper slightly to allow sliding the wrench towards the stem center. To engage, the gripper is moved such that the wrench head is rotating around the stem center. The gripper stays slightly opened during this motion to allow engaging of the wrench and prevent jamming due to induced forces from the manipulator. A successful engagement can be detected by a torque peak measured by the last joint (see Fig. 8). After engaging, the gripper is closed again for the subsequent valve turning.

7.2 Valve Rotation

We have to make sure to stay engaged during a full 360 ° rotation of the valve. For this purpose the manipulator applies a constant force of 10 N to push the wrench towards the stem center during valve rotation. Additionally, a torque of approximately 5 Nm is required, according to the competition specifications, to operate the valve stem. The resulting desired end-effector force $\mathbf{F}_{\text{ee}}$ is directly applied as feed-forward torque $\tau_{\mathbf{F}_{\text{ee}}} = \mathbf{J}_{\text{ee}}(\mathbf{q})^{\top}\mathbf{F}_{\text{ee}}$, where $\mathbf{J}_{\text{ee}}(\mathbf{q})$ is the end-effector Jacobian. Fig. 9 shows the commanded and measured torque of the manipulator's last joint during valve stem operation with a rotation speed of $0.25\,\text{rad}\,\text{s}^{-1}$. Noticeably, the actual required torque for rotation is less than 1 Nm.

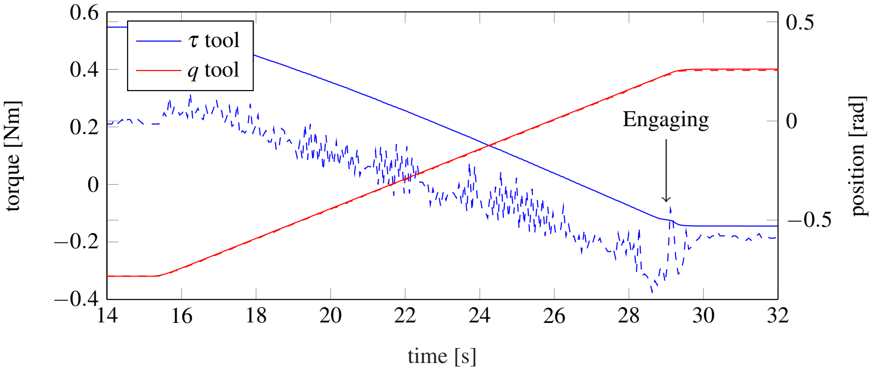


Fig. 8 Torque τ and position q tracking during valve engaging for the manipulator's last joint (tool). The solid lines show the commanded state, the dashed line the actual joint position and torque.

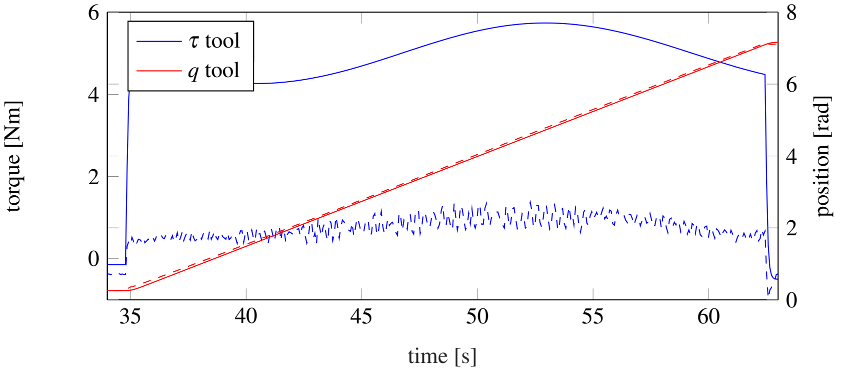


Fig. 9 Torque τ and position q tracking during valve rotation for the manipulator's last joint. The desired end-effector torque (solid) corresponds to the specified 5 Nm resistance, but the actual required torque (dashed) corresponds to less than 1 Nm.

8 Grand Challenge Performance

Both trials during the Grand Challenge showed similar performance. The second attempt lasted approximately 327 s of which 90 s were needed for exploration and navigation until positioned in front of the wrench panel, 95 s for measuring the valve stem and selecting the correct wrench, 110 s for grasping the wrench and engaging it with the valve, and the remaining 32 s for the actual rotation of the valve stem. The main bottleneck during navigation was the slow average driving speed of approximately 0.3 m/s which was owed to the limited turning rate necessary to avoid stick-slip effects between wheels and ground. During manipulation, the end-effector motion was required to be slow and steady for reliable and accurate visual tracking. Especially during grasping, slow approaching was necessary because of the shaking wrenches due to wind gusts.

A common failure source was the loss of target tracking during visual servoing because of changing lighting conditions. To recover from this case the arm moved to the last reference pose and restarted the tracking. Particularly important for successful manipulation is the precise positioning in front of the panel, as the arm with its limited range has to reach both the valve stem and the wrenches. We plan to overcome this limitation in the future through combined arm-base motion.

Overall, we successfully demonstrated the applicability of our system design by accomplishing the full valve operation task during both trials of the grand challenge of MBZIRC. Our mobile manipulator demonstrated its ability to operate completely autonomously even in the presence of network dropouts and adverse environmental conditions (high temperature, wind). Our modular approach proved valuable by being fast to adapt to a new environment, allowing software faults to be isolated, and making replacements of mechanical components easy. We will continue to use this system for further applications in real-world scenarios and as a research platform.

Acknowledgements This work was supported in part by the Swiss National Science Foundation (SNF) and the National Centre of Competence in Research Digital Fabrication. The authors would like to thank Sven Beck, Fabian Günther, and Eris Sako for their support in hardware design and Dario Bellicoso for his helpful advice.

References

1. Bodie, K., Bellicoso, C.D., Hutter, M.: ANYpulator: Design and control of a safe robotic arm. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1119–1125. IEEE (2016)
2. Calisti, M., Cianchetti, M., Manti, M., Corucci, F., Laschi, C.: Contest-driven soft-robotics boost: the RoboSoft Grand Challenge. Frontiers in Robotics and AI **3**, 55 (2016)
3. Corke, P.I.: Visual control of robot manipulators – a review. Visual servoing **7**, 1–31 (1993)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 (2005)
5. Davies, E.: Machine Vision: Theory, Algorithms, and Practicalities. Elsevier, New York (1997)
6. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
7. Flandin, G., Chaumette, F., Marchand, E.: Eye-in-hand/eye-to-hand cooperation for visual servoing. In: IEEE International Conference on Robotics and Automation (ICRA), vol. 3, pp. 2741–2746 (2000)
8. Guizzo, E., Ackerman, E.: The hard lessons of darpa's robotics challenge [news]. IEEE Spectrum **52**(8), 11–13 (2015)
9. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Schölkopf, B.: Support vector machines. IEEE Intelligent Systems and their Applications **13**(4), 18–28 (1998)
10. Hutter, M., Gehring, C., Jud, D., Lauber, A., Bellicoso, C.D., Tsounis, V., Hwangbo, J., Bodie, K., Fankhauser, P., Bloesch, M., Diethelm, R., Bachmann, S., Melzer, A., Hoepflinger, M.: ANYmal - a highly mobile and dynamic quadrupedal robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 38–44 (2016)
11. Joachims, T.: Making Large-Scale SVM Learning Practical. MIT Press, Cambridge, MA, USA (1999)
12. Kitano, H., Asada, M., Noda, I., Matsubara, H.: RoboCup: Robot world cup. IEEE Robotics & Automation Magazine **5**(3), 30–36 (1998)
13. Lima, P.U., Nardi, D., Kraetzschmar, G., Berghofer, J., Matteucci, M., Buchanan, G.: RoCKIn innovation through robot competitions. IEEE Robotics & Automation Magazine **21**(2), 8–12 (2014)
14. Pomerleau, F., Colas, F., Siegwart, R., Magnenat, S.: Comparing ICP variants on real-world data sets. Autonomous Robots **34**(3), 133–148 (2013)
15. Pomerleau, F., Krüsi, P., Colas, F., Furgale, P., Siegwart, R.: Long-term 3D map maintenance in dynamic environments. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 3712–3719 (2014)
16. Riener, R.: The Cybathlon promotes the development of assistive technology for people with physical disabilities. Journal of neuroengineering and rehabilitation **13**(1), 49 (2016)
17. Schneider, F.E., Wildermuth, D., Wolf, H.L.: ELROB and EURATHLON: Improving search & rescue robotics through real-world robot competitions. In: International Workshop on Robot Motion and Control (RoMoCo), pp. 118–123. IEEE (2015)
18. Siciliano, B., Caccavale, F., Zwicker, E., Achtelik, M., Mansard, N., Borst, C., Achtelik, M., Jepsen, N.O., Awad, R., Bischoff, R.: EuRoC-The Challenge Initiative for European Robotics. In: International Symposium on Robotics (ISR/Robotik), pp. 1–7. VDE (2014)
19. Suzuki, S., et al.: Topological structural analysis of digitized binary images by border following. Computer vision, graphics, and image processing **30**(1), 32–46 (1985)