

Article

Towards Tethered Tool Manipulation Planning with the Help of a Tool Balancer [†]

Daniel Sanchez, Weiwei Wan *  and Kensuke Harada 

Graduate School of Engineering Science, Osaka University, Osaka 560-8531, Japan; sanchez@hlab.sys.es.osaka-u.ac.jp (D.S.); harada@sys.es.osaka-u.ac.jp (K.H.)

* Correspondence: wan@sys.es.osaka-u.ac.jp

[†] This paper is an extended version of our paper published in Sánchez, D., Wan, W., Harada, K. Arm Manipulation Planning of Tethered Tools with the Help of a Tool Balancer. In Proceedings of the IFToMM World Congress on Mechanism and Machine Science, Krakow, Poland, 15–18 July 2019; Springer: Cham, Switzerland, 2019; pp. 2567–2576.

Received: 29 January 2020; Accepted: 27 February 2020; Published: 6 March 2020



Abstract: Handling and maneuvering tools across a robot workspace is a challenging task that often requires the implementation of constrained motion planning. In the case of wired or tethered tools, their maneuvering becomes considerably harder by the tool cable. If the cable presence is not considered, the robot motions may make the cable become entangled with the robot arms or elements of its workspace, causing accidents or unnecessary strain on the robot and the tool. Furthermore, the behavior of the tool cable during manipulation and its degree of entanglement around the robot are difficult to predict. The present paper introduces a constrained manipulation planner for dual-armed tethered tool manipulation involving tool re-grasping. Our solution employs a tool balancer to straighten the tool cable and facilitate the cable deformation problem. The planner predicts the cable states during manipulation and restricts the robot motions in order to avoid cable entanglements and collisions while performing tool re-posing tasks. Furthermore, the planner also applies orientational constraints to limit the cable bending, reducing the torque and stress suffered by the robot due to the cable tension. Simulations and real-world experiments validated the presented method.

Keywords: motion planning; object grasping; wired tools

1. Introduction

Recently, the inclusion of robots in the manufactory industry has reduced human workloads, increased productivity, and reduced costs in many industries. However, industrial robots are often found in enclosed and controlled environments with minimal or null human presence. In industrial scenarios, a robot usually has to handle tools to execute different tasks. In those cases, the tool's inherent characteristics such as its weight, shape, and cabling represent hindrances for tool manipulation. A tool's cable, in particular, represents a difficult problem for robotic manipulation: the robot might become entangled with the cable, causing damage to the tool or itself, or it might excessively bend the cable causing wear and tear or its breakage, as shown in Figure 1.

This paper presents a manipulation planning solution for the manipulation of tethered tools. The proposed planner uses an entanglement quantification method to estimate the degree of entanglement between the robot and a tool cable. The planner keeps track of the changes in cable orientation, and its relation to the grasping angle of the tool to compute the maximum rotation the cable could have around the robot arms. By keeping this rotation under a given threshold, the planner minimizes robot–cable entanglements and reduces the strain caused by the cable on the robot joints. Furthermore, the planner also limits the bending of the cable during manipulation, reducing the torque

caused by the cable on the robot joints. Accordingly, we review the related publications, including those addressing motion planning and manipulation planning.

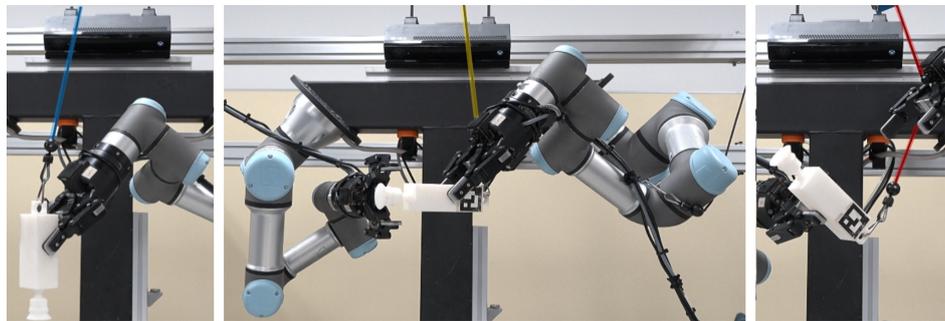


Figure 1. Cable entanglement-related accident: A robot performs a regrasp manipulation sequence with a tethered tool. The planned sequence does not address the tool cable. As a consequence, the cable becomes entangled around the robot arm causing excess strain on the robot. Finally, the robot also pulls from the cable, modifying the in-hand position of the tool, causing the manipulation task to fail. The light colors on the table indicate different status: blue, free; yellow, bent; red, entangled.

1.1. Motion Planning

Motion planning is a widely researched topic in robotics. Seminal work on motion planning include path planning algorithms [1], techniques for establishing lower bounds in robot motion planning problems [2], motion planning with fuzzy logic [3,4], the use of neural networks for planning [5,6], and the inclusion of genetic algorithms in motion planning algorithms [7–9].

More recently, newer solutions for motion planning have been presented. In [10], a data-driven approach for motion planning is proposed. A demonstration-guided motion planning framework was developed by Ye and Alterovitz [11]. In [12], a robot-specific circuitry for motion planning building process is presented.

In the field of motion planning, maneuvering and manipulating objects represents a difficult challenge. Manipulation planning is often considered to be a constrained case of motion planning—the manipulated object represents a dynamic obstacle that the robot must avoid. The motion sequences generated by manipulation planning allow a robot to move objects and to modify its environment’s structure.

1.2. Manipulation Planning

Manipulation consists in finding a series of intermediate object poses to pick-up an object in a given starting pose and re-pose it in a goal pose. Early work in manipulation solutions for robotic arms [13–15] explore regrasp planning in fixed single-armed robots. These work present some guidelines for future research in manipulation planning.

More recently, multiple works on manipulation planning have been presented. These work include single- and dual-arm algorithms for regrasping [16–18], a motion planning architecture on a hardware-in-the-loop application [19], a planner that computes robot configurations to grasp assembly parts for sequences of collaborative assembly operations [20], and 3D object reorientation using pivoting [21].

On the other hand, robot manipulation planning involving cable-like objects has presented many challenges; these objects are difficult to manipulate, with often unpredictable deformations and, in the case of cables, they can become entangled around the robot or its surroundings. Several works centered on manipulation planning have tackled different problems associated with deformable objects such as cables, chains, hoses, etc. For example, Hert et al. [22] developed a planner for multiple tether robots in 2D space, where the tails are treated as lines or bounded segments. Igarashi et al. [23] also presented a 2D planner for a mobile robot with a tail. The tails are treated as a manifold in the

configuration space. Bretl et al. [24] approximated flexible wires locally using a chart in the 3D world and demonstrated its usage in quasi-static manipulation planning. Ramirez et al. [25] developed a finite element model for a fire hose and used it to generate the pulling motion of a humanoid robot. In [26], a framework for cable shapes manipulation using a deformation model is shown. A motion planner for manipulating deformable linear objects and tying knots was described by Saha and Isto [27], while Khalil and Payeur [28] reviewed the works on dexterous robotic manipulation of deformable objects by using sensor feedback.

1.3. Contribution

The aforementioned works on object manipulation suggest general solutions for object reorientation, but the wires of tools are overlooked. It remains difficult to include cable shape prediction, collision, and entanglement avoidance as well as motion planning in the same loop. This paper uses a tool balancer to convert the cable constraints into a straight line, bypassing the necessity of modeling the cable to predict its behavior. It also implements a constrained planner that addresses the problems caused by wired cables using movement constraints. The constraints limit excessive bending of the cable, reducing stress on the robot joints. They also help the robot avoid becoming entangled with the tool cable as well as avoid accidents related to cable snarling. This paper is an extension and continuation of our previous work [29].

2. Materials and Methods

2.1. Real-world Setting

Our robot environment simulates an industrial workspace. Figure 2 shows the robot handling a suspended tool. The tool is attached to a tool balancer, a device designed to suspend tools and aid factory workers by reducing the load of carrying a tool. The tool balancer provides a constant pulling force that straightens the cable. The robot counts with hand-mounted cameras used to detect AR markers placed on a tool. The object initial pose is calculated from the AR marker position and orientation. We use our simulation environment along with our manipulation planner to plan a robot manipulation sequence to place the object in a desired pose.



Figure 2. (left) The robot workspace. Our robot consists of two UR3 robotic arms attached to a metal frame; the robot uses its hand-mounted cameras (light blue) to detect the position of a 3D printed tool. The tool possesses an AR marker (blue square) for pose detection. The tool is suspended by a tool balancer (purple). (right) (a.1–2) Front and back view of a tool balancer. (b.1) The cable of the tool balancer presents a constant pulling force that simplifies the cable deformation into a straight line.

2.2. Simulation Setting

The simulation environment (https://gitlab.com/wanweiwei07/wrs_nedo) is used to simulate robot–object–environment interactions and plan robot manipulation motion sequences (Figure 3). The simulation environment uses 3D meshes to represent the tool and the tool balancer in the robot environment. The tool original pose is determined with the robot hand-mounted cameras and the tool AR marker [30].

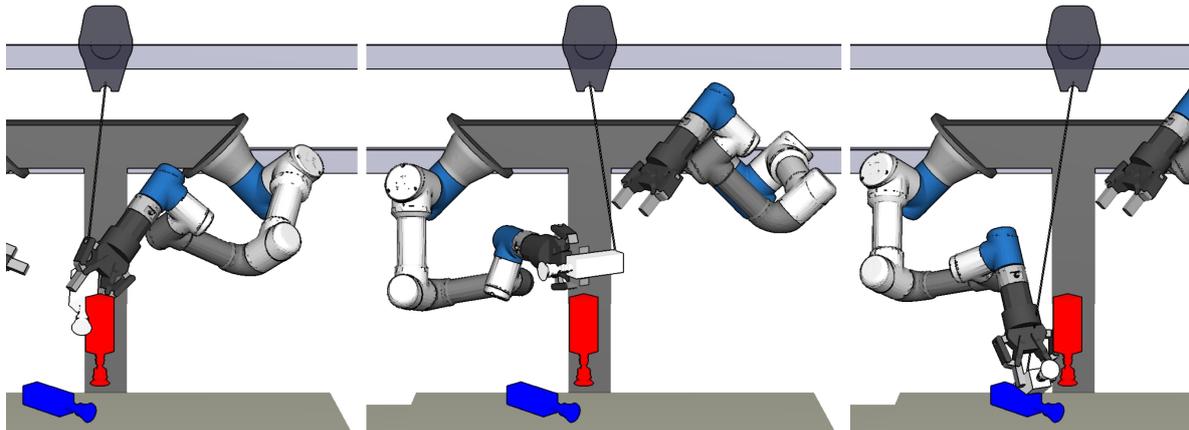


Figure 3. A simulation of the robot performing a tool re-posing task using regrasp. **(left)** The robot grasps and moves the tool from its original pose (red) using its left arm. **(middle)** The robot performs a handover to grasp the object with its right arm. **(right)** The robot moves the object to its goal pose (blue). The simulation places the tool balancer and represents the tool cable as a straight line that goes from the tool's back to the tool balancer.

2.3. Manipulation Planner

Our planner [16] uses a pre-built database of grasps, object placements, and handover pairs of grasps to build a graph of grasps. When we give a start and goal object pose to the planner, it searches the grasp graph to find IK-feasible poses associated to the initial and final grasps of the object.

For handover execution, the planner also searches the graph to find pairs of grasps that are compatible with the starting and final grasp poses. If a compatible set of initial grasp–handover–final grasp is found, the planner uses RRT to connect these key poses and create a motion sequence for the robot to complete its task. It is also possible for the planner to find solutions that involve more than one handover operations where the object moves from hand to hand several times, which involves more key poses. Naturally, there is often more than one solution or set of compatible pick up–handover–place down grasps to complete the task. In that case, the planner chooses randomly between the options that involve the fewest key poses. In the event that a complete motion sequence cannot be found for a given set of key poses (due to unavoidable collisions in the motion sequence or an inability for the robot to preserve its balance), the planner selects another set of key poses and starts the RRT exploration again.

The generated motion sequence allows the robot to re-pose a tool in a desired goal pose. However, in the case of tethered tools, it is necessary to impose constraints on the planning stage to limit the bending of the cable and avoid entanglements with the robot or its environment. The following sections discuss these constraints. They are added to our planner developed in [16] to ensure neither arm becomes entangled. The dual-arm planner is sequential: the constraints are added to each single arm, respectively.

3. Constraints for Entanglement Avoidance

To prevent entanglements around the robot, we have implemented two constraint-based solutions to our planner. The first set of constraints, presented in [29], uses orientation constraints to prevent the robot from bending the tool cable past a certain threshold, reducing the torque suffered by the robot

during manipulation. Our second set of constraints makes use of a proposed method to measure the possible snarling or “angle accumulation” of the cable around the robot arm. In this case, the planner imposes a maximum threshold on angle accumulation, avoiding entanglements during manipulations. Both methods were implemented in our planner.

3.1. Orientation Constraints

Our planner uses orientational constraints to limit the bending of the tool cable around the tool itself during the manipulation of the tool. For every robot–tool state calculated during sampling-based motion generation, the planner measures the cable bending. If the bending surpasses a given threshold, the planner discards said poses and searches a new pose that complies with the threshold. The process is repeated until a complete motion path is found. The planner uses two vectors α° and β° in the robot reference frame Σ_o to measure the bending of the cable. α° is a reference vector and β° represents the vector formed by the cable straight line shape.

3.1.1. Computation of the Bending Angles

To calculate the bending of the cable for a given robot–tool pose, the planner first computes the vectors α° and β° . For any given tool pose, the planner calculates the vector α° by inserting the vector ${}^t v = {}^t z$ into Equation (1).

$${}^\circ v = {}^\circ \mathbf{R}_t {}^t v \quad (1)$$

where ${}^\circ \mathbf{R}_t$ is the object’s rotational matrix, ${}^t v$ is a given vector in the tool’s reference frame Σ_t , and ${}^\circ v$ represents the same vector in the robot’s frame of reference Σ_o . In our case, we chose ${}^t v = {}^t z$ since the z-axis tool reference frame aligns with the tool connection point normal vector; the resulting vector ${}^\circ v$ is equal to the reference vector ${}^\circ \alpha$.

The vector β° is calculated by computing the straight line formed by the cable. The line starts at the tool connection point and ends at the tool balancer connection point, as shown in Figure 4.

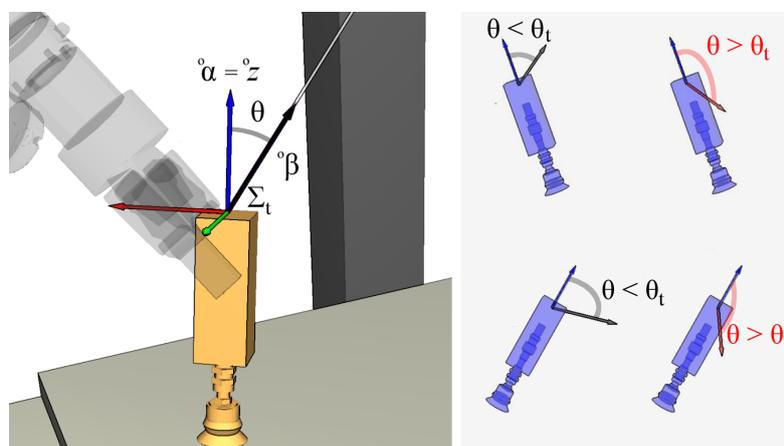


Figure 4. The left picture shows the robot arm grasping a tethered tool. The vector ${}^\circ \beta$ (black arrow) points to the cable (white line) direction and originates at the tool connection point. The vector ${}^\circ \alpha$ (blue arrow) is equal to the normal vector of the connection point surface, which, in this case, is equal to the z vector in the tool reference frame Σ_t . The right picture shows examples of cable bending that stay below a threshold $\theta_t = 90^\circ$ (in black) and cable bendings that surpass the threshold (red).

3.1.2. Angular Measurement of Cable Bending

Our planner uses vectors ${}^{\circ}\alpha$ and ${}^{\circ}\beta$ to measure the bending angle θ . The angle measurement is performed for every robot–tool state generated by the planner during the sampling-based motion generation (in our case, RRT). To compute the angular difference θ , the planner uses Equation (2)

$$\theta = \arccos \frac{{}^{\circ}\alpha \cdot {}^{\circ}\beta}{|{}^{\circ}\alpha| |{}^{\circ}\beta|} \quad (2)$$

to limit the bending of the cable, we set a maximum threshold, θ_t , for the value of θ . If a given robot–tool state generated during RRT exploration has a value θ that surpasses θ_t , the planner considers the cable to be excessively bent, violating our constraint. In such cases, the planner discards the robot pose and continues searching for robot poses that do not violate the constraint until a complete motion path without excess cable bending is found. Figure 4 presents our simulation environment with different tool orientations and illustrates vectors ${}^{\circ}\alpha$ and ${}^{\circ}\beta$ and their angular difference.

3.2. Angle Accumulation Constraints

The orientation constraints offer a direct solution to limit the bending of the tool cable during manipulation. However, the angle constraints do not directly prevent cable entanglements. To address cable entanglements, we use a method to predict the magnitude of cable “snarling” or cable movement around the robot arm during simulation [31]. By measuring the angle accumulation of every robot state during the planning stage and preventing the accumulation from surpassing a given threshold, it is possible to create entanglement-free motion sequences for tool manipulation.

The concept of angle accumulation [31] represents the degree of rotation or movement of the tool cable around the robot arm. For simplicity, our planner uses polar coordinates in the tool reference frame to quantify the movement of the cable. The angle accumulation is measured by computing the azimuthal and polar rotations of the cable within an “accumulation region”. The accumulation region is determined by the polar angle used by the robot hand to grasp the tool ϕ_H and the azimuthal positioning of the robot hand, as shown in Figure 5.

When the cable enters the accumulation region (red region in the figure) the planner considers the cable to be in a state of angle accumulation and starts recording the changes in polar and azimuthal angles of the cable to determine if the movements of the cable contribute to get itself entangled around the robot arm or the tool. Our planner can make use of the angle accumulation estimation to inspect individual robot–cable estates/poses during the planning stage. Similar to the orientation constraints, we can introduce a threshold or maximum permissible angle accumulation to limit the snarling and movement of the cable around the robot arms.

In principle, it is not always possible to calculate the angle accumulation for a single robot state since it depends on the preceding accumulation values and the direction of rotation of the cable. Our planner calculates each state angle accumulation during RRT-based motion planning to create entanglement-free motion paths from an initial robot key pose/state and a goal.

The i th robot–cable state in a motion sequence has an angle accumulation $\Gamma(i)$ that depends on the previous states directions of rotation (clockwise or counter clockwise) of the cable. The accumulation $\Gamma(i)$ is calculated by comparing the changes of orientation of the β vector with Equation (3). It is important to notice that β_0 represents the first cable state that crosses the accumulation region.

$$\Gamma(i) = \sum_{j=1}^i ((\Theta(\beta_j) - \Theta(\beta_{j-1}))\mu(\beta_{0,1,\dots,j}) + (\phi(\beta_j) - \phi(\beta_{j-1}))\Phi(\beta_{0,1,\dots,j})) \quad (3)$$

where the functions $\Theta(\beta)$ and $\phi(\beta_j)$ return the azimuthal and polar angles of the i th cable vector β_i , respectively. For simplicity, the cable vector β uses the tool reference frame unlike the vector ${}^{\circ}\beta$ discussed in the previous subsection; otherwise, both vectors represent the orientation of the cable. The functions $\mu(\beta_{0,1,\dots,j}), \Phi(\beta_{0,1,\dots,j}) \in \{-1, 1\}$ are used to change the sign of the subtraction. Both

functions can be seen as memory functions that verify if the change in cable state from β_i to β_{i+1} contributes to increasing the accumulation or decreasing it. The functions check if the net contribution from previous cable states results in a clockwise or counterclockwise rotation accumulation and, if necessary, it changes the sign of the operations to increase or decrease the polar or azimuthal accumulation accordingly.

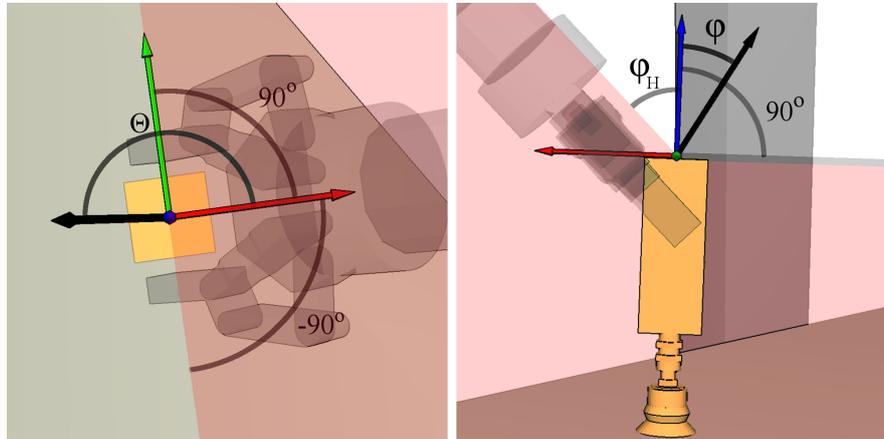


Figure 5. Sideways views of the robot holding a tethered tool. The tool reference frame, represented by the red, green and blue arrows (x, y and z axes in the tool reference frame) are used to measure the polar and azimuthal angles of the cable, represented by ϕ and Θ respectively. The grasp used by the robot determines the entanglement region (in red): the orientation with which the robot holds the tool ϕ_H (right) and the octants in which the hand is located (left) define the region. If the cable vector (black arrow) crosses the entanglement (red) region, the planner counts how much the cable rotates in said region. The total rotation is called angle accumulation.

Angle Accumulation Constraints in Sampling-based Motion Planning

To keep track of how the angle accumulates during sampling-based motion planning, we use unidirectional RRT to create nodes/states (robot/object intermediate poses) that connect the starting tool pose to the goal tool pose. The planner stores the orientation of the cable for every robot state β_i as a vector in the tool reference frame and its accumulation angle. To verify the angle accumulation of a given node, the planner compares the β vector of the node with its parent node. The planner reads the direction of rotation (clockwise or counterclockwise) accumulation in the polar and azimuthal planes of the parent node and uses Equation (3) to compute the angle accumulation.

If the angle accumulation for a particular state/node surpasses a given threshold, the state's node is eliminated from the permissible nodes and a new node is created. The process continues until a set of connected nodes that link the start and goal pose is found or the planning stage times out. In the latter case, a new set of starting and goal poses is searched and the RRT exploration is restarted. Figure 6 better illustrates this process. The figure shows two hypothetical motion paths generated by RRT: The first path allows the robot to take the object from a starting pose and place it in a desired goal pose without surpassing angle accumulation thresholds. The second path makes the robot reach the same goal pose at the cost of high angle accumulation, which may cause entanglements.

The threshold Γ_t is used to limit the movement of the cable in the entanglement zone and prevent entanglements. Essentially, by limiting the movement of the cable in the accumulation region shown in Figure 5, we decrease the chance of the cable snarling around the robot arm or colliding with the robot end-effector during handover, as shown in Figure 1.

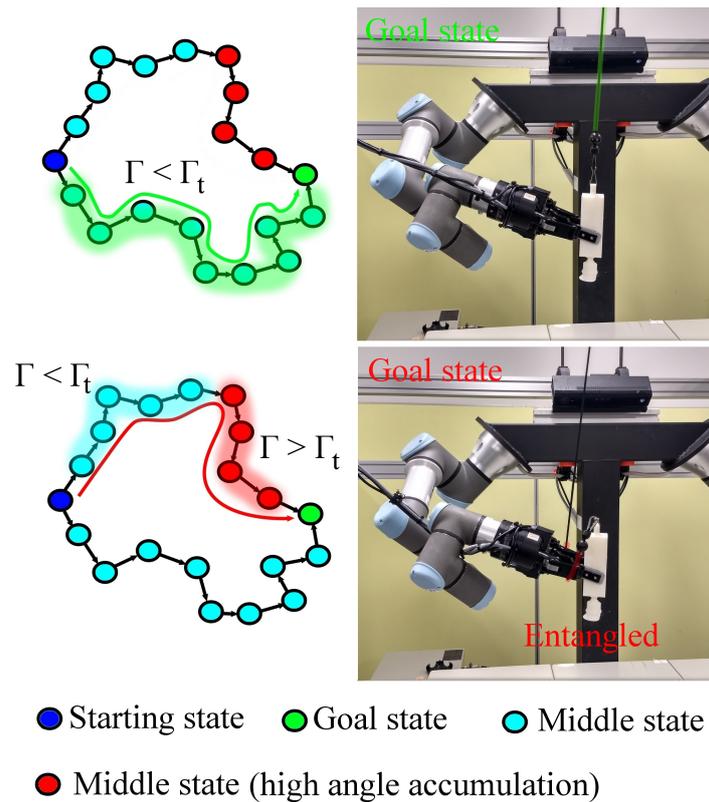


Figure 6. The figure shows an RRT-generated tree with starting (blue) and goal (green) nodes that represent a robot state or its joint angle configuration (R^6). The RRT exploration generates two possible paths composed of intermediate robot states (represented by the light blue and red nodes) that connect the start and goal nodes. The path with red nodes makes the cable enter a state of high angle accumulation during the manipulation task, causing the cable to get snarled by the time the robot reaches its goal joints (lower right image). The other path, chosen by our planner, makes the robot reach the same goal pose without incurring entanglements. Our planner avoids motion paths that make the tool cable incur a high angle accumulation state ($\Gamma > \Gamma_t$) to avoid cable entanglements.

4. Experiments

Several experiments were performed to test and compare our proposed planner to our original planner [16]. The tests involved different object reposing tasks with distinct starting and goal poses. Simulations and real-world experiments are presented. For the simulations, we provide a running time comparison between the presented solution and our original planner and a comparison between the planners' rate of success in computing manipulation motion sequences to solve the given tasks. For the real-world experiments, the robot force/torque sensor was used to measure the torque suffered by the robot during manipulation. In addition, we measured the angle accumulation of the cable to compare the possible degree of snarling caused by the tool cable between both planner solutions.

4.1. Simulations

To assess the performance of our constrained planner and compare it to the original, we performed a series of tests in our simulation environment. Firstly, we tested the planners' success rates by tasking the robot to perform several manipulation tasks with a tethered tool. We designed several benchmarks to test our planner. Each benchmark consists of generating a motion sequence to grasp the tool with the robot left arm, perform a handover, and place the tool in one of the five goal poses shown in Figure 7. We tested our proposed planner using $\theta_t = 120^\circ$ as threshold for the angle bending and $\Gamma_t = 90$ for the angle accumulation threshold and compared it to our original motion planner. The results of the tests can be seen in Table 1. To calculate the planner's success rate with different manipulation tasks, we

labeled the execution of the tasks using the following criterion: If the tool's bending angle calculated with Equation (2) was higher than the given threshold angle at any point of the task, the execution was labeled as failed by excessive bending of the cable; if the angle accumulation surpassed Γ_r , the task was identified as a failed case due to entanglement with the cable.

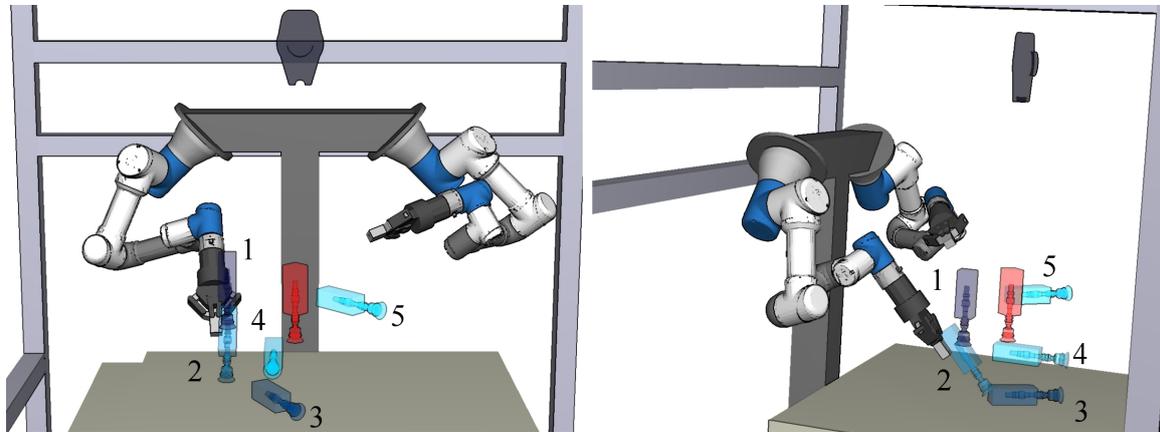


Figure 7. The figure shows the different goal poses given to the planner (the tool in different shades of blue) and the starting pose of the object (in red). For these experiments, the robot must grasp the object with its left arm, perform a handover operation, and place the tool in a given goal pose (1–5) or a combination of goal poses (for example, first go to Pose 5 and then Goal 2).

Table 1. Simulation results for the constrained and unconstrained planner.

Benchmark	Proposed Planner	Unconstrained Planner
1	○{82.65}	×{57.68}
2	○{87.57}	× / * {63.66}
3	○{80.19}	*{59.38}
4	○{85.40}	○{64.24}
5	○{80.80}	× / * {69.24}
5,2	⊗	× / * {67.63}
1,3	○{196.66}	×{76.91}
4,3	○{180.11}	× / * {84.52}

Symbols: ○ is used to represent successful attempts to pose the object in the goal pose without excess bending or angle accumulation. × indicates that the cable was bent above our threshold angle of 120° . * indicates that the angle accumulation surpassed the threshold $\Gamma = 90^\circ$. ⊗ indicates that the planner could not find a solution within the maximum planning time. The planning time for each completed benchmark is shown inside { } with the time in seconds. The benchmark number indicates which tool goal pose or poses the planner was tasked to place the tool in.

4.2. Real-world Experiments

After performing the simulations, we tested the planned motion sequences in our real-world robot. The motion sequences computed by our proposed planner can limit the cable bending and the movement of the cable around the robot arm. However, it is necessary to test the sequences in a real-world environment to assess how well the implemented constraints or thresholds in cable bending and angle accumulation translate into motion sequences that produce less torque on the robot arms and avoid cable snarling around the robot.

Therefore, we tasked our real-world robot to complete several of the benchmarks presented in the previous section by using our proposed planner and our original unconstrained planner.

Figure 8 shows some of the robot executions. To assess the planners ability to prevent entanglements, we registered the number of entanglement accidents caused by cable snarling. Every accident consists of the cable getting attached around the robot arm and the robot undesirably pulling the cable when performing a handover motion. The upper row of Figure 8 shows an example of an entanglement-related accident. Furthermore, we registered the torque suffered by the robot during the manipulation tasks. Table 2 shows the results gathered from the experiments, including the registered cable-related accidents during manipulation as well as the mean and maximum torque values registered by the force/torque sensor of the robot UR arms. Figure 9 shows the torque measurements for each benchmark. With the constraints, the total torque resisted by the robot is reduced (see blue vs. green; red vs. yellow). The peaks appear when the robot switches between planned motion segments. The blue and red curves had more peaks than the green and yellow ones, which implies that the constraints increased the complexity of the planned motion. This is consistent with our expectation that the robots are leveraging the motion changes to avoid entanglement. Note that the constraints do not necessarily increase the planning time—the blue and red curves are not always longer than the green and yellow ones. The planning time depends on the obstacle manifolds in the configuration space. In some cases, the constraints guide the robot to circumvent configuration obstacles and the planning time decreases.

Table 2. Real-world results for the constrained and unconstrained planner.

Benchmark	Proposed Planner	Unconstrained Planner
1,3	○ [0.25, 0.87]	⊗ [0.84, 1.25]
2	○ [0.63, 0.75]	⊗ [1.27, 1.57]
3	○ [0.69, 1.01]	○ [1.66, 1.03]
4,3	○ [0.95, 0.68]	⊗ [2.28, 2.43]
5	○ [0.83, 1.09]	⊗ [1.41, 1.56]

Symbols: ○ is used to represent successful attempts to pose the object in the goal pose without entanglement-related accidents. ⊗ indicates that the robot suffered an accident during manipulation caused by the cable. Inside [] we show the average torque [$N \cdot m$] caused by the cable on the robot end-effector right and left arms, respectively.

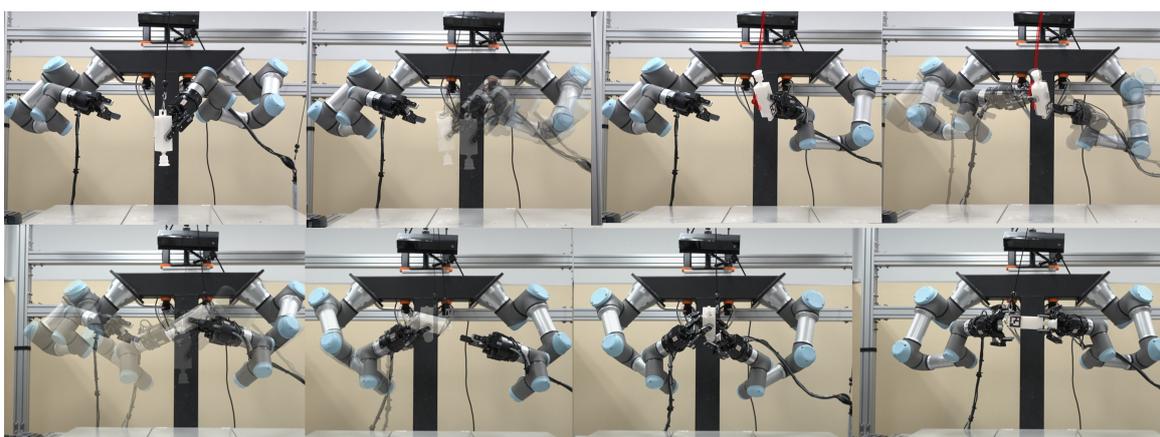


Figure 8. Comparison between the constrained and unconstrained planner. The upper motion sequence, computed with the unconstrained planner, shows the robot getting entangled with the tool cable (red), making it fail the manipulation task. The lower sequence, computed by our proposed planner, shows the robot taking an alternative motion path to preserve angle accumulation and cable bending below the given thresholds. The result is a complete motion sequence without entanglement-related accidents or excess cable bending.

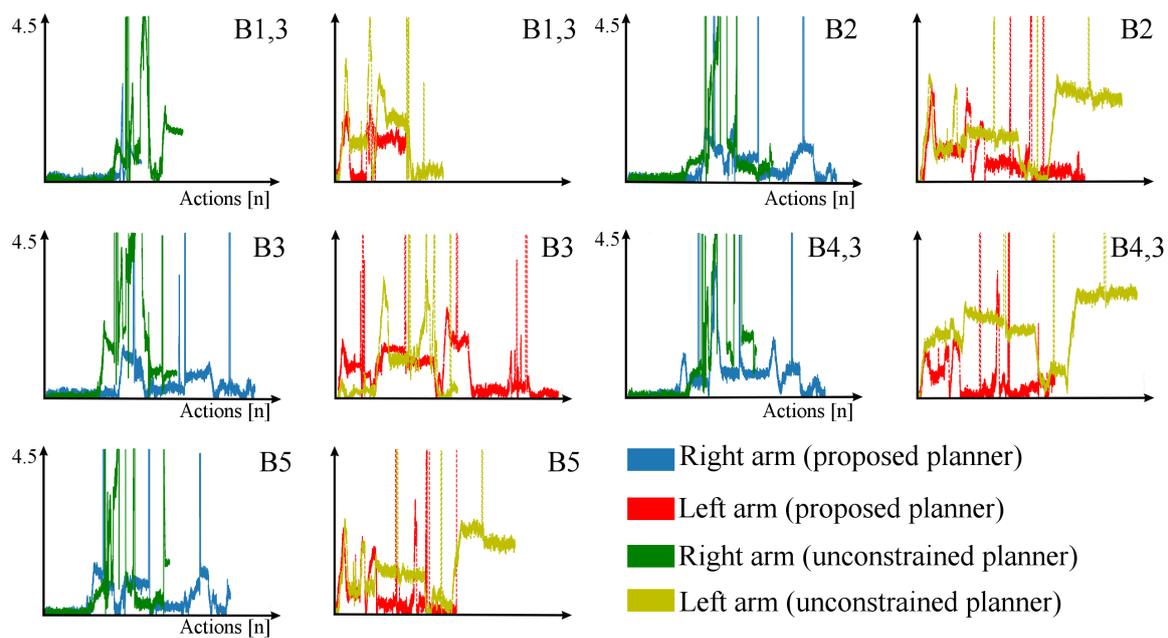


Figure 9. Comparison between the constrained and unconstrained planner. Horizontal axis shows actions (sequence ids); Vertical axis shows torques ($[N \cdot m]$). With the constraints, the total torque resisted by the robot due to the torque of the balancer's cable is reduced. Experimental results show that the torque applied by the cable is lower when the planner uses the proposed constraints, effectively reducing the strain on the robot joints.

5. Discussion

A constrained planner for tethered tool manipulation is presented. The planner limits the bending angle of the tool cable and constrains the cable movement in the vicinity of the robot arms, successfully reducing stress caused by the cable tension as well as avoiding entanglements with the cable. Limiting the bending of the cable can reduce the torque suffered by the robot arms during manipulation and decrease entanglements. With the unconstrained planner, the average torques suffered by the robot arms were $1.56N \cdot m$ and $1.492N \cdot m$ for the right and left arms, respectively; our planner reduced the average torques to $0.88N \cdot m$ and $0.67N \cdot m$, respectively, thus highly reducing the torque suffered by the robot. However, the imposed threshold constraints can reduce the search space for key poses and limit the available robot configurations during planning, considerably increasing the average planning time when compared to the unconstrained planner. Results confirm that the planner can successfully compute motion sequences for robots to handle balancer-suspended tools, preventing collisions with a tool's cable and its excess bending, thus reducing the strain suffered by the robot by diminishing the torque applied by the balancer's retractable cable on its joints. The planner is expected to play promising roles in manufacturing cells.

Author Contributions: Conceptualization, D.S. and W.W.; Data curation, D.S.; Formal analysis, D.S.; Funding acquisition, W.W. and K.H.; Investigation, D.S.; Methodology, D.S., W.W. and K.H.; Project administration, W.W. and K.H.; Resources, W.W. and K.H.; Software, D.S. and W.W.; Supervision, W.W.; Validation, D.S.; Visualization, D.S. and W.W.; Writing and original draft, D.S.; Writing and review and editing, W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is based on results obtained from a project commissioned by the New Energy and Industrial 269 Technology Development Organisation (NEDO).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lozano-Pérez, T.; Wesley, M.A. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM* **1979**, *22*, 560–570. [\[CrossRef\]](#)
2. Canny, J.; Reif, J. New lower bound techniques for robot motion planning problems. In Proceedings of the 28th Annual Symposium on Foundations of Computer Science (sfcs 1987), Los Angeles, CA, USA, 12–14 October 1987; pp. 49–60.
3. Shibata, T.; Fukuda, T. Intelligent motion planning by genetic algorithm with fuzzy critic. In Proceedings of the 8th IEEE International Symposium on Intelligent Control, Chicago, IL, USA, 25–27 August 1993; pp. 565–570.
4. Vachtsevanos, G.; Hexmoor, H. A fuzzy logic approach to robotic path planning with obstacle avoidance. In Proceedings of the 25th IEEE Conference on Decision and Control, Athens, Greece, 10–12 December 1986; pp. 1262–1264.
5. Zacksenhouse, M.; DeFigueiredo, R.J.; Johnson, D.H. A neural network architecture for cue-based motion planning. In Proceedings of the 27th IEEE Conference on Decision and Control, Austin, TX, USA, 7–9 December 1988; pp. 324–327.
6. Janglová, D. Neural networks in mobile robot motion. *Int. J. Adv. Robot. Syst.* **2004**, *1*, 2. [\[CrossRef\]](#)
7. Sugihara, K.; Smith, J. Genetic algorithms for adaptive motion planning of an autonomous mobile robot. In Proceedings the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation', Monterey, CA, USA, 10–11 July 1997; pp. 138–143.
8. Gen, M.; Cheng, R.; Wang, D. Genetic algorithms for solving shortest path problems. In Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97), Indianapolis, IN, USA, 13–16 April 1997; pp. 401–406.
9. Chen, M.; Zalzalá, A.M. A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration. *J. Robot. Syst.* **1997**, *14*, 529–544. [\[CrossRef\]](#)
10. Pfeiffer, M.; Schaeuble, M.; Nieto, J.; Siegwart, R.; Cadena, C. From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. In Proceedings of the 2017 IEEE international conference on robotics and automation (icra), Marina Bay Sands, Singapore, 29 May–3 June 2017; pp. 1527–1533.
11. Ye, G.; Alterovitz, R. Guided motion planning. In *Robotics Research*; Springer: Berlin, Germany 2017; pp. 291–307.
12. Murray, S.; Floyd-Jones, W.; Qi, Y.; Sorin, D.J.; Konidaris, G. Robot Motion Planning on a Chip. In Proceedings of the Robotics: Science and Systems, Ann Arbor, MI, USA, 18–22 June 2016.
13. Tournassoud, P.; Lozano-Pérez, T.; Mazer, E. Regrasping. In Proceedings of the 1987 IEEE International Conference on Robotics and Automation, Raleigh, NC, USA, 31 March–3 April 1987; pp. 1924–1928.
14. Stoeter, S.A.; Voss, S.; Papanikolopoulos, N.P.; Mosemann, H. Planning of regrasp operations. In Proceedings of the 1999 IEEE International Conference on Robotics and Automation, Detroit, MI, USA, 10–15 May 1999; pp. 245–250.
15. Lozano-Pérez, T.; Jones, J.L.; O'Donnell, P.A.; Mazer, E. *Handey: A Robot Task Planner*; MIT Press: Cambridge, MA, USA, 1992.
16. Wan, W.; Harada, K. Developing and comparing single-arm and dual-arm regrasp. *IEEE Robot. Autom. Lett.* **2016**, *1*, 243–250. [\[CrossRef\]](#)
17. Calandra, R.; Owens, A.; Jayaraman, D.; Lin, J.; Yuan, W.; Malik, J.; Adelson, E.H.; Levine, S. More Than a Feeling: Learning to Grasp and Regrasp using Vision and Touch. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3300–3307. [\[CrossRef\]](#)
18. Wan, W.; Harada, K. Reorientating objects with a gripping hand and a table surface. In Proceedings of the 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), Seoul, Korea, 3–5 November 2015; pp. 101–106.
19. La Mura, F.; Todeschini, G.; Giberti, H. High performance motion-planner architecture for hardware-in-the-loop system based on position-based-admittance-control. *Robotics* **2018**, *7*, 8. [\[CrossRef\]](#)
20. Dogar, M.; Spielberg, A.; Baker, S.; Rus, D. Multi-robot grasp planning for sequential assembly operations. *Auton. Robot.* **2019**, *43*, 649–664. [\[CrossRef\]](#)

21. Hou, Y.; Jia, Z.; Mason, M.T. Fast Planning for 3D Any-Pose-Reorienting Using Pivoting. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 1631–1638.
22. Hert, S.; Lumelsky, V. The ties that bind: Motion planning for multiple tethered robots. *Auton. Robot.* **1996**, *17*, 187–215. [[CrossRef](#)]
23. Igarashi, T.; Stilman, M. Homotopic path planning on manifolds for cabled mobile robots. In *Algorithmic Foundations of Robotics IX*; Springer: Berlin, Germany, 2010; pp. 1–18.
24. Bretl, T.; McCarthy, Z. Quasi-static manipulation of a Kirchhoff elastic rod based on a geometric analysis of equilibrium configurations. *Int. J. Robot. Res.* **2014**, *33*, 48–68. [[CrossRef](#)]
25. Ramirez-Alpizar, I.G.; Naveau, M.; Benazeth, C.; Stasse, O.; Laumond, J.P.; Harada, K.; Yoshida, E. Motion generation for pulling a fire hose by a humanoid robot. In Proceedings of the 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, Mexico, 15–17 November 2016; pp. 1016–1021.
26. Zhu, J.; Navarro, B.; Fraise, P.; Crosnier, A.; Cherubini, A. Dual-arm robotic manipulation of flexible cables. In Proceedings of the IROS: International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018.
27. Saha, M.; Isto, P. Motion planning for robotic manipulation of deformable linear objects. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 2478–2484.
28. Khalil, F.F.; Payeur, P. Dexterous robotic manipulation of deformable objects with multi-sensory feedback—a review. In *Robot Manipulators Trends and Development*; InTechOpen: London, UK, 2010.
29. Sánchez, D.; Wan, W.; Harada, K. Arm manipulation planning of tethered tools with the help of a tool balancer. In *IFTToMM World Congress on Mechanism and Machine Science*. Springer: Berlin, Germany, 2019; pp. 2567–2576.
30. Raessa, M.; Sánchez, D.; Wan, W.; Petit, D.; Harada, K. Teaching a robot to use electric tools with regrasp planning. *CAAI Trans. Intell. Technol.* **2019**, *4*, 54–63. [[CrossRef](#)]
31. Sanchez, D.; Wan, W.; Harada, K. Tethered Tool Manipulation Planning with Cable Maneuvering. *arXiv* **2019**, arXiv:1909.10686.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).