In-Hand Manipulation Using Three-Stages Open Loop Pivoting

Silvia Cruciani and Christian Smith

***Abstract*— In this paper we propose a method for pivoting an object held by a parallel gripper, without requiring accurate dynamical models or advanced hardware. Our solution uses the motion of the robot arm for generating inertial forces to move the object. It also controls the rotational friction at the pivoting point by commanding a desired distance to the gripper's fingers. This method relies neither on fast and precise tracking systems to obtain the position of the tool, nor on real-time and high-frequency controllable robotic grippers to quickly adjust the finger distance. We demonstrate the efficacy of our method by applying it on a Baxter robot.**

I. Introduction

Many tasks in robotics require object interaction and tool use, whereby the robot picks up and grasps these in a way that is suitable for the overall task. However, the desired grasp may be difficult or unfeasible to achieve from the initial position. Moreover, even when the robot can plan correctly, the resulting grasp may be different from the desired one due to uncertainties in the environment and imprecise motion execution. As a consequence, the robot needs to reposition the object to be able to execute its final task.

One common approach for repositioning uses regrasping with *pick-and-place*. In this approach, the robot places the object on a surface and plans a different grasp to pick it up again in the desired configuration [1]. Such a method requires a space next to the robot for the placing action and it also needs time to execute.

Other approaches mimic the human ability to change grasp configuration by moving the fingers precisely, exploiting the *intrinsic dexterity* of the hand. These methods provide an efficient solution for in-hand regrasping. However, replicating a human-like dexterity on a robotic hand requires a gripper with high mechanical complexity and good coordination of the multiple degrees of freedom to achieve an in-hand regrasping [2], [3]. Alternatively, the dexterity of the robot can be enhanced by designing customized grippers specifically for regrasping [4], but this introduces the need for a specific kind of hardware to achieve a desired task.

Many robots have simple parallel grippers that are robust and easy to control, but poor in intrinsic dexterity. To compensate for this lack of degrees of freedom, and perform in-hand manipulation with a simple gripper, a robot can use *extrinsic dexterity*. This approach takes advantage of external supports such as gravity, contact points and inertial forces [5]–[7].

In this paper, we focus on a specific kind of repositioning strategy, called pivoting. This strategy consists of rotating the object (or the tool) between two fingers to reorient it with a desired angle, as illustrated in Fig. 1. Pivoting is a robust method for in-hand manipulation that allows for reorientation along a single axis. This reorientation is enough for many tasks, and it can also be combined with sliding actions [8] for obtaining a wider range of repositioning.

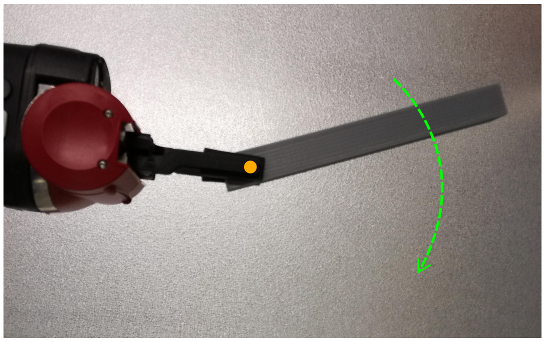

Fig. 1: An example of a pivoting task in which the tool, held by a parallel gripper, has to rotate around a pivoting point, marked in orange.

An example of a task that requires pivoting consists of a robot that is holding a screwdriver and needs to use it. For most robotic manipulators, to turn a screw the screwdriver has to be positioned close to 0° with respect to the gripper. However, when picking up the screwdriver from a table, this orientation can be unfeasible to achieve. Therefore, the tool has to pivot within the end-effector.

Most of the proposed solutions for pivoting require both a high-frequency control of the gripper to rapidly adjust the finger distance (the distance between the two jaws of the gripper), and thereby the friction, and the ability to track the orientation of the tool while it moves at high velocity. However, many commercial robots and grippers do not provide a real-time or high-frequency control of the gripper's fingers (e.g. Baxter, Yumi). Moreover, it is difficult to obtain a precise and robust vision-based tracking system, especially without using a high frame-rate camera, due to motion blur and susceptibility to changes in illumination and the object's appearance.

In this paper, we propose a method for the pivoting task that also satisfies the following conditions:

- Low hardware requirements. The method requires neither a high-precision and high-frequency controllable robotic manipulator, nor a high-speed tracking system to determine the orientation of the tool.
- Low modeling needs. This method does not rely on high-precision modeling; it uses a simple mathematical model and a rough measure of the involved parameters.

Silvia Cruciani and Christian Smith are with the Robotics, Perception and Learning Lab, CSC at KTH Royal Institute of Technology, Stockholm, Sweden. {cruciani, ccs}@kth.se

Furthermore, the proposed method is suitable to be used while the robot is performing other tasks, since it does not pose hard constraints on the velocity of the robot's end-effector, but it only requires a limited set of conditions to be satisfied.

The proposed method performs pivoting in three distinct stages:

1) in the first stage, the gripper that is holding the tool starts moving until it reaches a desired velocity;
2) in the second stage, the gripper stops and opens the fingers at a desired distance, to allow the tool to rotate;
3) in the third stage, the tool rotates around the pivoting point until in reaches the desired angle.

During the third stage, no feedback is required, as no action to influence the motion of the tool is performed. Consequently, the rotation of the tool is solely determined by its initial angular velocity, by the friction at the pivoting point and by the gravity. The initial angular velocity depends on the gripper's speed reached during the first stage, and the friction at the pivoting point depends on the finger distance reached during the second stage.

We consider the initial angular velocity and the finger distance as the inputs of the system, and we compute the optimal values using Q-learning.

While the lack of feedback control reduces the controllability of the tool's rotational motion, it increases the possibility of application of our method. This allows us to satisfy all the conditions for working with simple hardware setups, as we do not require high-frequency control to adjust the finger distance or high-speed tracking systems. Our method pivots an object using open loop control to lower the requirements on cameras used for tracking. In addition, the gripper's fingers are used to influence the motion of the tool only at the beginning of the pivoting action, without further precise readjustments. Hence, there is no need for high-frequency control of the gripper.

II. Related Work

Previous works on pivoting involve environmental constraints such as external contact surfaces, motions of the robot arm that generate inertial forces to produce angular momentum, and external forces such as gravity.

In [9], the authors propose to use a contact surface to rotate an object between two stable poses. In this case, there is no control on the gripping force.

On the other hand, several works on pivoting strongly focus on controlling the force applied on the object by the gripper's fingers. By controlling this force it is possible to change the torsional friction that influences the object's motion. In [10] the authors focus on *swing-up* motions and they exploit the ability of the gripper to exert dissipative torque on the object. The proposed solution uses an energy-based control that pivots the object to the angle that has the desired potential energy. They extend the discussion in [11], in which they synthesize the approach to regrasp an object from a lower energy angle to a higher one. In this approach, the motion of the object is limited to the vertical plane. Moreover, it strongly relies on fast response time in controlling the gripper and on fast sensory feedback to track the position of the object at every time-step.

The adaptive control solution proposed in [12] exploits the gravity acceleration and the friction at the pivoting point. This work has successively been extended by including tactile sensors to measure the normal force applied by the gripper in [13]. In this approach, the gripper does not move and the motion of the object depends on the direction of gravity. Therefore, it is only possible to change the object's configuration toward a position with lower potential energy than the initial one. In addition, the gripper's fingers need to be controlled at a high frequency to adjust their distance and the object needs to be tracked precisely during the entire process.

These works rely on high-frequency control of the gripper and on precise and fast sensory feedback to track the motion of the object over time. Moreover, the motion appears to be constrained on the vertical plane and does not take into account possible changes of orientation of the plane of rotation.

Unlike other methods, the one that we propose requires neither a high-frequency real-time controllable gripper, nor a high-speed tracking system to determine the orientation of the tool. In fact, we want to achieve a successful pivoting on a commercial robot, using an easily available camera to determine the orientation of the tool with respect to the gripper. Our method uses a low-frequency controllable parallel gripper and only requires measurements of the angle between the tool and the gripper when they are not moving, working with still images. Moreover, the proposed method is generalizable to cover variations in the plane of rotation of the tool.

III. Problem Formulation

This section provides a formalization of the problem and it describes the model of the system.

The system is composed by a tool held by a parallel gripper, which is the end-effector of a robot arm that can be velocity controlled in 3D space.

We assume that we can approximately measure the dimensions of the grasped tool and its mass, or infer them from the available sensors. To be generally applicable to a wide range of robots, we assume we cannot directly measure the force applied to the tool at the pivoting point. This force influences the torsional friction. We also assume that the coefficients that describe this friction are unknown or highly uncertain.

Because we use modest commercial hardware, we assume that the execution of the commands is subject to errors. In particular, we assume errors in the finger distance execution and in the gripper's velocity, which are due to errors in the robotic manipulator's joints actuation.

A. Pivoting Task

Given a tool held by a parallel gripper with an angle θ_0 with respect to the gripper itself, the goal is to rotate the tool until it reaches the desired angle θ_d. More specifically, we want to find the necessary initial velocity $\dot{\theta}_0$ that the gripper has to transmit to the tool and the necessary finger distance at which the parallel gripper needs to open in order to obtain a motion that ends at the desired final angle.

B. Sliding Friction and Deformation Model

Once the tool has started moving, the relevant forces affecting its motion are gravity and torsional friction at the pivoting point. The torsional friction depends on the opening of the fingers: a wide opening gives less friction and a narrow opening gives more friction.

When the tool is not moving, we use the Coloumb model to describe the static friction τ_s as:

$$|\tau_s| \leq \gamma f_n, \tag{1}$$

in which γ is the static friction coefficient and f_n is the normal force applied by the fingertips on the tool at the pivoting point.

We use viscous and Coulomb friction [14] to model the torsional friction τ_f when the tool is moving:

$$\tau_f = -\mu\dot{\theta} - \sigma\,\mathrm{sgn}(\dot{\theta}) f_n, \tag{2}$$

where μ and σ are the viscous and Coulomb friction coefficients respectively, $\dot{\theta}$ is the angular velocity of the tool around the pivoting point and sgn is the signum function.

Since our system cannot measure f_n directly, we follow the approach suggested in [12] and we express it as a function of the finger distance d using a linear deformation model:

$$f_n(t) = k(d_0 - d), \tag{3}$$

where k is a stiffness parameter and d_0 is the distance at which the fingers initiate contact with the tool, i.e. the distance of zero deformation for the fingertips.

With $\xi = \sigma k$, we write the overall torsional friction as:

$$\tau_f = -\mu\dot{\theta} - \xi\,\mathrm{sgn}(\dot{\theta})(d_0 - d). \tag{4}$$

To avoid numerical singularities when the tool starts moving, or, in other words, when it initiates the switching between the two models in Equation 1 and 4, we follow the approach suggested in [15]: we define a small neighbor $|\dot{\theta}| \leq \epsilon$ (for a small $\epsilon > 0$) in which the normal force on the tool counterbalances the net torque to preserve equilibrium.

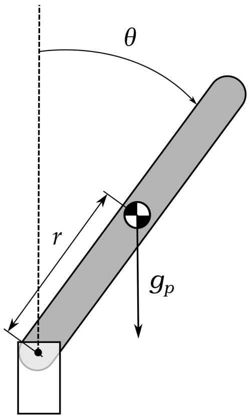

Fig. 2: A tool that rotates around a fixed pivoting point.

C. Dynamic Model

Our system is composed by a tool rotating around a pivoting point, as shown in Fig. 2. The pivoting point corresponds to the contact point between the tool and the fingers of the robotic gripper that is holding it. Since there is no additional force influencing the motion of the tool, its dynamics are determined by the gravity acceleration and the torsional friction, as:

$$(I + mr^2)\ddot{\theta} - mg_p r \sin(\theta) = \tau_f, \tag{5}$$

in which I is the inertia of the tool with respect to its center of mass, m is its mass, r is the distance between its center of mass and the pivoting point and g_p is the component of the gravity acceleration in the plane of rotation of the object.

While the tool is moving the gripper is still. Therefore, the direction of the gravity component in the plane of rotation remains constant.

The parallel gripper is the end effector of a robotic manipulator, which can change its orientation in space. Depending on the orientation, g_p can vary between $-9.8\ \mathrm{m/s^2}$ and $9.8\ \mathrm{m/s^2}$. As shown in Fig. 2, the direction of gravity corresponds to the axis at which $\theta = 0$. However, it is trivial to generalize it to other situations by including the difference in angle between these two directions.

D. Error Analysis

We assume that the actuation of the gripper's motion is subject to errors. More specifically, we assume that the executed velocity will be different from the desired one, and that the actual finger distance will also slightly differ from the commanded one.

The error in velocity execution affects the estimate of the initial angular velocity of the tool, which affects the estimate of the torsional friction. We express the actual angular velocity $\tilde{\dot{\theta}}$ as:

$$\tilde{\dot{\theta}} = \alpha\dot{\theta}_d, \tag{6}$$

in which $\dot{\theta}_d$ is the desired angular velocity and $\alpha \geq 0$ is a coefficient that describes the error. In particular, $\alpha < 1$ describes a slower motion than the one desired and $\alpha > 1$ describes a faster one. By using a friction coefficient $\tilde{\mu} = \mu\alpha$ in Eq. 4, this error can be incorporated in the estimate of the friction coefficients.

The error in the finger distance affects the estimate of the normal force at the pivoting point, hence the torsional friction. This error can be absorbed in the friction coefficient ξ similarly to the error in the tool's angular velocity.

This formulation compensates for errors in actuations by translating them into error in the friction coefficients' values. These values are estimated in order to adapt to the behavior of the real system.

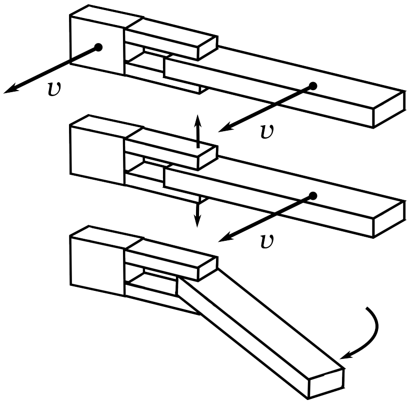

Fig. 3: The three separate stages of the open loop pivoting. In the first stage, the gripper and the tool move at the same velocity. In the second stage, the gripper stops and opens the fingers. In the last stage, the tool rotates until it reaches the desired angle.

```
Algorithm 1: three-stages pivoting
Input: initial estimate of $\mu$, $\xi$, target angle $\theta_d$, initial angle $\theta$
while $|\theta_d - \theta| > \delta$ do
    compute optimal control action $a^*$
    execute stage 1
    execute stage 2
    read new angle $\theta$
    update $\mu$, $\xi$
end
```

IV. Three-Stages Pivoting

We propose a method for pivoting that is composed of three separate stages, as shown in Fig. 3. This approach needs neither a fast and reliable tracking system nor a high-frequency control of the gripper's fingers, nor a highly precise estimate of the parameters describing the friction. In fact, these coefficients can be estimated on-line, i.e. while the robot is manipulating the tool. Whenever these parameters are updated, the necessary commands to achieve a desired pivoting actions will change consequently.

A detailed description of the three stages and of the evaluation of the necessary control actions is presented below, while a summary of our method is shown in Algorithm 1.

A. The Three Stages

It is necessary to control the motion of the robotic manipulator that is holding the tool in order to generate the desired motion and reach the target angle. The specific method for determining the desired initial velocity $\dot{\theta}_0^*$ of the tool and the desired finger distance d^* is explained in section IV-B.

Our method is divided into three separate stages:

1) End-effector's velocity stage: in this stage, the gripper holds the tool firmly, and it moves until it reaches a desired velocity. This desired velocity is such that, as soon as the tool starts rotating in the third stage, its initial angular velocity at the pivoting point will be $\dot{\theta}_0 = \dot{\theta}_0^*$, which combined with the proper finger distance will allow the tool to reach the desired angle. In the instant in which the tool starts moving, while the gripper stops, its center of mass continues its motion with the previous velocity $\mathbf{v}$. Since the motion of the tool is constrained, this velocity corresponds to an angular velocity ω around the pivoting point. This angular velocity is:

$$\omega = \frac{\mathbf{v} \cdot \hat{\mathbf{r}}_\perp}{r}, \tag{7}$$

in which $\hat{\mathbf{r}}_\perp$ is the unitary vector orthogonal to the vector that goes from the pivoting point to the tool's center of mass and $\cdot$ is the scalar product. Hence, the desired velocity $\mathbf{v}$ of the gripper is the one so that $\omega = \dot{\theta}_0^*$.

With this method, the robotic manipulator is free to move the gripper in any direction and at any velocity, as long as this constraint is satisfied for the successful outcome of a pivoting action. This is useful when the pivoting action is combined with another motion of the robot to accomplish a higher level task. For instance, the robot can plan a trajectory for tool use so that when the tool stops in front of the object to interact with, it pivots to the required orientation, imposing no additional constraints to the arm motion.

Since the synchronization between the end-effector's stop and the fingers' opening is subject to errors, the velocity that is transferred to the tool will have small alterations. However, this error can be included in the estimation of the parameters as shown in section III-D.

2) Finger distance stage: In this stage, the end-effector stops and opens the fingers at a desired distance. This distance is d^*, which corresponds to the desired torsional friction at the pivoting point.

The friction coefficients μ and ξ may typically not be known a priori for a new tool, and they also have to be estimated. This estimation is run in parallel to the execution of the pivoting task, and the desired actions can be recomputed accordingly whenever a change in the coefficients is estimated. This allows us to include an adaptation of the method using observations from the real system. Once the estimate of these parameters improves, the robot will be able to reliably pivot the tool to any desired angle. More information on the parameters' update is provided in section IV-C.

3) Tool's motion stage: In this stage, the tool rotates around the pivoting point until it reaches the desired angle. Its motion is solely determined by the actions taken in the previous stages.

Due to the influence of gravity, it is possible that once the tool stops it soon starts rotating in the opposite direction. To avoid this undesired motion, which would lead the tool in a wrong configuration, the gripper's fingers can be closed to firmly grasp the tool at the time in which it reaches the desired angle. This time can be computed from the solution of Eq. 5. It is easy to obtain a numerical solution to this differential equation, given $d = d^*$ and the initial condition $\dot{\theta}_0 = \dot{\theta}_0^*$. Alternatively, the possible values of d can be restricted so that the static friction, once the tool stops, prevents successive motions due to gravity, without the need of closing the fingers.

B. Selection of the Control Action

The formula in Eq. 5 depends on the current state of the tool, described by θ and $\dot{\theta}$, and by the current finger distance d, which influences the torsional friction τ_f. Since we open the gripper at the beginning, the distance d does not vary while the tool moves. Therefore, the differential equation only depends on θ. It is possible to solve it numerically by imposing a condition on the initial velocity $\dot{\theta}_0$.

Moreover, since the tool keeps moving in the same direction, it is possible to determine the value of $\text{sgn}(\dot{\theta})$ by looking at the starting and desired angle. Alternatively, it is also possible to assume this sign to always be positive and change the reference system for the angles accordingly every time. This allows us to avoid possible numerical discontinuities.

Once the tool starts moving, it stops at a final angle θ_f. This final angle depends on the initial velocity $\dot{\theta}_0$ and on the finger distance d. These two variables correspond to the first two stages of our method. On a real setup, these variables are limited by the robot's joints velocities, by their precision and by the accuracy of the gripper in adjusting the finger distance.

While it is possible to compute the exact inputs required to reach a desired angle given an initial one, this computation would not provide any solution if the final angle were not reachable in a single step. In fact, the maximum opening of the fingers and the maximum achievable velocity impose a limit to the maximum angle that can be reached. Moreover, assuming low accuracy of the robot hardware, it may not be able to reliably execute a commanded velocity at the end-effector, and the position of the tool is also subject to possible errors in the detection. Hence, we prefer to use a discretization of the control variables and of the states in order to introduce a margin of tolerance to these potential errors.

From how we defined it, this problem is easily solvable using Reinforcement Learning algorithms or Dynamic Programming. This allows us to learn the best control inputs to execute in order to obtain the desired angle, and it can easily generalize to many initial angles at the same time. Among the possible choices, we choose to use a Q-Learning approach to "learn" the proper action [16], but many other approaches will work.

The control action a to learn is composed by the initial desired velocity and by the desired finger distance:

$$a = (\dot{\theta}_0, d). \tag{8}$$

The tool's angle θ represents the state of the system. We use the following reward function R:

$$R(\theta) = \begin{cases} 1 & \text{if } |\theta - \theta_d| \leq \delta \\ 0 & \text{otherwise} \end{cases}, \tag{9}$$

in which δ represents the margin of tolerance for the goal.

The learning process is executed without interaction with the real system. It predicts the outcome of an action in a given state by using the model in Eq. 5. With the Q-learning's discount factor, the actions that are preferred are the ones that lead the tool to the goal using a single control action, when possible.

It is possible to learn continuous actions from a continuous representation of spaces. However, the precision obtained from learning continuous values for $\dot{\theta}_0$ and d requires also precision in the actuation of the robotic manipulator to execute exactly those values. Moreover, we want the learning process to be as fast as possible because we propose to use it while the robot is currently manipulating a tool in case the parameters are uncertain. Therefore, as already mentioned, we use a discretization of actions and states. Given our state and action, this discretization leads to a small number of possible states and actions that does not slow down the process of learning and allows the execution of the actions to have a tolerance margin.

After the learning, given a state θ of the tool we are able to determine an optimal action $a^* = (\dot{\theta}_0^*, d^*)$. This action is one of the best combinations of $\dot{\theta}_0$ and d that allows the tool to reach the desired angle. In case a new tool is grasped, the process of learning will be repeated accordingly to new estimate in the friction coefficients.

C. Parameters Update

Unlike the mass and length of the tool, which are easily obtainable, the friction coefficients μ and ξ of a new tool are unknown at the beginning. Unless some previous stage of estimation is performed, the actions learned using default or highly uncertain parameters will not produce any meaningful result when executed on the robot.

We propose to update the friction coefficients while the robot is trying to execute a pivoting task. The robot executes the actions learned using default coefficients or using a rough estimate of them if it is available. The outcome of these actions is not successful, because the tool does not stop at the desired angle. Therefore, the parameters can be adapted to match the observed behavior.

In particular, the friction coefficients are estimated to minimize the error:

$$e = \sum_{i=1}^{N} (\theta_{exp,i} - \theta_{obs,i})^2, \tag{10}$$

in which θ_{obs} is the observed angle, i.e. the real outcome of the action, and θ_{exp} is the expected outcome of the actions given the known friction coefficients. This last variable can be estimated by obtaining a numerical solution from Eq. 5, knowing the initial angular velocity and the finger distance from the executed action. N is the total number of attempts to reach the desired angle executed so far.

Once the friction coefficients are estimated correctly, the new optimal actions are able to pivot the tool to the desired angle. Successively, by using the estimated friction coefficients, it is possible to learn how to reach different angles in a single step, and to manipulate the tool as desired.

V. Experiments

We performed several experiments to test our proposed approach for pivoting.

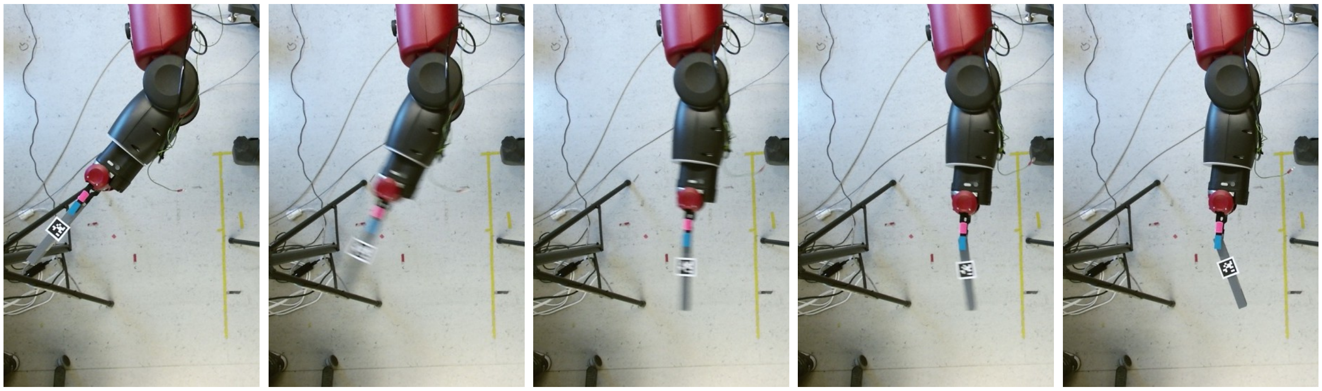

Fig. 4: Pivoting task executed with the Baxter robot's gripper. The first image shows the tool at an angle $\theta_0 = 0$. The three following images show the three stages of our method: the first stage, in which the gripper and the tool are moving with the same velocity; the second stage, in which the gripper stops once the desired velocity for the tool has been reached, and the fingers open at the desired distance; the third stage, in which the tool rotates around the pivoting point. The last image shows the tool at the desired angle $\theta_d = 0.52$ radians.

$I[\mathrm{kg} \cdot \mathrm{m}^2]$	0.000057248
$m[\mathrm{kg}]$	0.024
$r[\mathrm{m}]$	0.084
$d_0[\mathrm{m}]$	0.0189

TABLE I: Parameters of the tool used in the experiments

First, we demonstrated the performance of the three-stages controller, using pre-estimated values for the friction coefficients to isolate the controller from the parameters estimation. We tried to reach two different target angles from the same initial angle by manually resetting the tool whenever the goal was reached. Then, to show the generalization to different orientations of the end-effector, we tried to reach a third angle with a different orientation of the plane of rotation. Fig. 4 shows one of these experiments.

In the second set of trials, we assessed the performance when the system is commanded to reach an angle without first assigning correct values to the parameters, so that our method needs to estimate them on-line, while trying to reposition the tool.

A. General Setup

These experiments were performed on a Baxter robot. The parallel gripper had slightly deformable fingertips to be able to modify and control the friction according to the commanded finger distance.

The parameters of the tool we used are shown in table I. The tool's shape was rectangular. However, our method is generalizable to different object shapes, because it depends only on an estimate of the inertia and of the center of mass. Moreover, it is generalizable to different materials thanks to the on-line friction coefficients estimation.

To determine the angle of the tool with respect to the gripper, we used an AprilTag [17] placed on the tool, and used a Kinect 2 for detection.

For the Q-learning implementation, we used PyBrain [18]. We set an epsilon-greedy explorer for the learning. The angle was reset to the initial one at the beginning of every episode and whenever the tool hit the goal. Fig. 5 shows an example of the learning for this problem when the target angle was set to 0.52 radians. This learning was sufficiently fast (an episode of 50 iterations run on average for 18 ms) to be re-computed online in case of a re-evaluation of the friction coefficients. Therefore, the process of estimating these parameters can be run in parallel to the execution of the actions and the learned actions will change according to the updated friction coefficients.

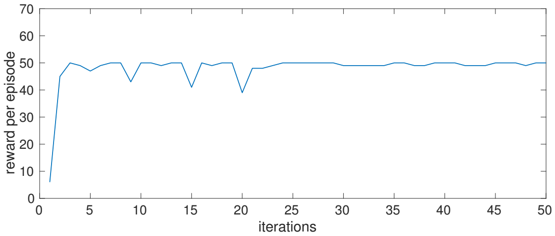

Fig. 5: The learning process as reward received per episode with the increasing number of iterations. The target angle is set to $\theta_d = 0.52$ radians.

During the first experiments, the tool was manually replaced in the gripper at a starting angle close to 0. Since we placed the tool manually, the position of the pivoting point on the tool varied slihgtly between trials, implying a variation in the value of r. We verified that a variation of the pivoting point within ≈ 1.0 cm does not affect the successful outcome of the execution.

Furthermore, we observed a behavior that sometimes affected the final outcome of an action: before stopping completely, the tool had a small bounce due to a non-modeled effect that would cause a small backward motion. We assume that this motion is caused by the deformation of the rubber material used for the fingertip. This motion is usually negligible, but it can be compensated by closing the fingers immediately instead of waiting for the tool to stop a second time.

B. Performances of the Three-Stages Control

For the first experiment we set 0.52 radians ($\approx 30°$) as the target angle. The actions were learned using friction coefficients that we had estimated beforehand as $\xi = 11.976$

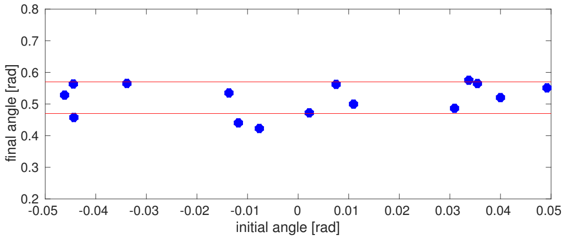

Fig. 6: A sample of the experiment from a neighbor of $\theta_0 = 0$ as initial state to $\theta_d = 0.52$ radians as goal state. The blue dots represent the outcome of the experiments and the red lines delimit the margins of the tolerated goal region.

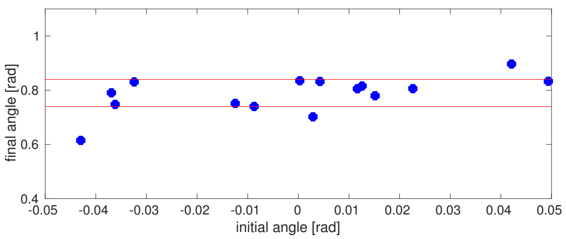

Fig. 7: A sample of the experiment from a neighbor of $\theta_0 = 0$ as initial state to $\theta_d = 0.79$ radians as goal state. The blue dots represent the outcome of the experiments and the red lines delimit the margins of the tolerated goal region.

and $\mu = 0.00568$. These estimates were accurate enough, so the learned actions did not need to adjust to further changes.

We set the rotational plane to be horizontal, so that the gravity acceleration was close to 0. This allowed us to verify that the angle at which the tool stopped was close to the expected one, without the need of closing the fingers immediately to compensate for the gravity acceleration. Moreover, it showed that the motion of the tool was caused only by the motion of the robot arm, therefore the influence of the gravity acceleration is not essential for pivoting.

We set the initial angle to 0°. The states ranged from 0 to 1.75 radians ($\approx 100°$) with a resolution of 0.05 radians ($\approx 3°$). This resolution allowed us to keep a goal tolerance that was small enough to obtain a final angle close to the desired one but it was also sufficiently large to compensate for small errors in the robot's actuation. The total number of states was 36. We set the goal tolerance δ to 0.1 radians.

The actions include both the finger distance and the initial velocity for the tool. Given the range of the fingers from firmly closed on the object to d_0 and the approximation in motion that we could obtain from the gripper's actuation, which we verified to have an accuracy of roughly 0.0005 m, we selected the following set as possible finger distances (in meters): $\{0.0171, 0.0175, 0.0179, 0.0183, 0.0187\}$. We limited the maximum initial velocity of the tool to 15 rad/s to keep a high safe margin from the robot's joint limits. We discretized it with steps of 0.2. Thus, the overall number of actions was 380.

Fig. 6 shows the result of our experiments. The image shows the different outcome of the experiments in a neighbor of the initial angle $\theta_0 = 0$. The unsuccessful experiments are still in a range of 0.097 radians ($\approx 5°$) from the target angle. Such variation is probably due to non-modeled behaviors that slightly affect the motion of the tool.

To show the usability of our method for different angles, we performed an experiment with a different target angle, $\theta_d = 0.79$ radians ($\approx 45°$). Since the starting angle was always the same ($\theta_0 = 0$), we noticed that the maximum velocity allowed for the tool was not sufficient to reach the desired angle in one step. Therefore, we increased the allowed velocity to 21.0 rad/s and lowered the resolution to keep the same number of actions. We verified that this increase in velocity was still in the range of actuation of the robot's joints, i.e. it would not lead to a violation of their velocity limits.

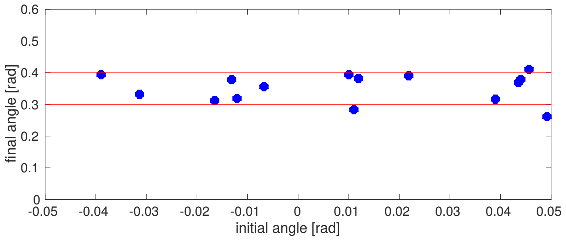

Fig. 8: A sample of the experiment from a neighbor of $\theta_0 = 0$ as initial state to $\theta_d = 0.35$ radians as goal state, with $g_p = -1.7$ m/s^2. The blue dots represent the outcome of the experiments and the red lines delimit the margins of the tolerated goal region.

We found that learning to reach this angle was more difficult than the previous one. Therefore, we experimented with different values of epsilon in the epsilon-greedy explorer to find the one that would allow a faster learning. Since, as already mentioned, the learning runs on-line while estimating the parameters during the experiments, it is important that this process is fast.

We performed the experiments on the real robot with the same tool and setup. Fig. 7 shows the outcome of these experiments. The first set of experiments is shown in the video attachment.

The last experiment of this set of trials involved a change in the orientation of the gripper, to show that the proposed method generalizes to variations in the plane of rotation of the tool. We set the end-effector's orientation so that $g_p = -1.7$ m/s^2. We verified that the static friction alone was enough to keep the tool in position once it stopped at the desired angle. This desired angle was set to $\theta_d = 0.35$ radians ($\approx 20°$) and the initial angle was $\theta_0 = 0$. We used the same set of actions of the previous experiment. The results are shown in Fig. 8.

C. *Performances of the Parameters Estimation*

Finally, we performed an experiment in which the initial coefficients were not correct for the right outcome of the learning at the beginning. The values we used were $\xi = 1.0$ and $\mu = 0.1$. These values were constantly updated during the attempts to reach the correct angle. We used pySMAC [19] to obtain an estimate that minimizes the error between the expected angle and the real one. We set the target angle to $\theta_d = 0.79$ radians and the initial angle to $\theta_0 = 0$. The

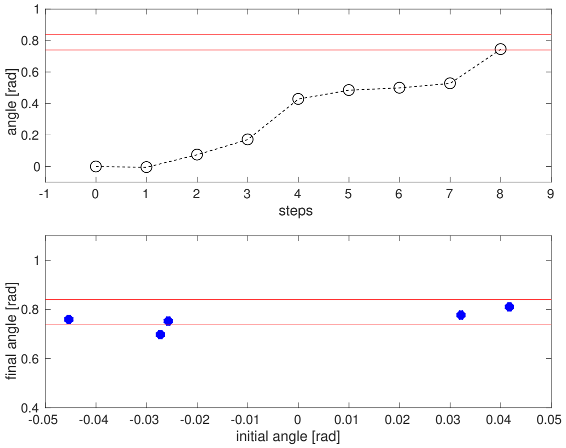

Fig. 9: The first image shows the steps before reaching the desired angle $\theta_d = 0.79$ radians from $\theta_0 = 0$. The black circles indicate the reached angle and the red lines delimit the goal region. The second image shows samples of trials after the parameters estimation, starting from a neighbor of $\theta_0 = 0$. The blue dots represent the obtained angle.

position of the tool was not manually reset until the tool reached the desired angle.

We used the same set of actions as in the previous experiment, and we set the plane of rotation to be horizontal. The process of reaching the desired angle took 8 steps, during which the friction coefficients were updated using the new observations. The goal was reached when these estimates were $\xi = 12.0131773$ and $\mu = 0.00496152$. To verify that these values are good enough for successive in-hand manipulations of the same tool, we repeated the experiment of reaching the target angle from an initial angle of $\theta_0 = 0$. This time, the angle was reached in one step. The process of reaching the target angle during the parameters estimation and a sample of the following experiments with the estimated parameters are shown in Fig. 9.

VI. Conclusions and Future Work

We proposed an approach for pivoting that allows a robot to successfully reorient an object held by a parallel gripper without the need for high-frequency and real-time controllable robotic grippers. We divided our approach in three separate stages and we used Q-learning to compute the action to perform during the first two stages. We showed the result of this approach by reorienting a tool to different target angles using a Baxter robot.

As future work, we plan to include this pivoting action in a more general task. For instance, the robotic manipulator has a tool to use, and this tool needs to be in a particular configuration in order to be used properly. We plan to develop a strategy for achieving the reorientation while performing other actions at the same time, leading to the successful outcome of the overall task.

Acknowledgment

This work was supported by the European Union framework program H2020-645403 RobDREAM.

References

[1] P. Tournassoud, T. Lozano-Perez, and E. Mazer, “Regrasping,” in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, Mar 1987, pp. 1924–1928.

[2] J. C. Trinkle and J. J. Hunter, “A framework for planning dexterous manipulation,” in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, Apr 1991, pp. 1245–1251 vol.2.

[3] N. Furukawa, A. Namiki, S. Taku, and M. Ishikawa, “Dynamic regrasping using a high-speed multifingered hand and a high-speed vision system,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, May 2006, pp. 181–187.

[4] N. Chavan-Dafle, M. T. Mason, H. Staab, G. Rossano, and A. Rodriguez, “A two-phase gripper to reorient and grasp,” in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2015, pp. 1249–1255.

[5] N. C. Dafle, A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge, “Extrinsic dexterity: In-hand manipulation with external forces,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 1578–1585.

[6] N. Chavan-Dafle and A. Rodriguez, “Prehensile pushing: In-hand manipulation with push-primitives,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 6215–6222.

[7] J. Shi, J. Z. Woodruff, and K. M. Lynch, “Dynamic in-hand sliding manipulation,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 870–877.

[8] R. D. Howe and M. R. Cutkosky, “Practical force-motion models for sliding manipulation,” *The International Journal of Robotics Research*, vol. 15, no. 6, pp. 557–572, 1996. [Online]. Available: http://ijr.sagepub.com/content/15/6/557.abstract

[9] A. Holladay, R. Paolini, and M. T. Mason, “A general framework for open-loop pivoting,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 3675–3681.

[10] A. Sintov and A. Shapiro, “Swing-up regrasping algorithm using energy control,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4888–4893.

[11] A. Sintov, O. Tslil, and A. Shapiro, “Robotic swing-up regrasping manipulation based on the impulse-momentum approach and clqr control,” *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1079–1090, Oct 2016.

[12] F. E. Viña, Y. Karayiannidis, K. Pauwels, C. Smith, and D. Kragic, “In-hand manipulation using gravity and controlled slip,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 5636–5641.

[13] F. E. Viña, Y. Karayiannidis, C. Smith, and D. Kragic, “Adaptive control for pivoting with visual and tactile feedback,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 399–406.

[14] H. Olsson, K. J. strm, M. Gfvert, C. C. D. Wit, and P. Lischinsky, “Friction models and friction compensation,” *Eur. J. Control*, p. 176, 1998.

[15] D. Karnopp, “Computer simulation of stick-slip friction in mechanical dynamic systems.” *J. Dyn. Syst. Meas. Control.*, vol. 107, no. 1, pp. 100–103, 1985.

[16] C. J. Watkins and P. Dayan, “Q-learning,” *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[17] E. Olson, “AprilTag: A robust and flexible visual fiducial system,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.

[18] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber, “PyBrain,” *Journal of Machine Learning Research*, vol. 11, pp. 743–746, 2010.

[19] F. Hutter, H. H. Hoos, and K. Leyton-Brown, “Sequential model-based optimization for general algorithm configuration,” in *Proc. of LION-5*, 2011, p. 507523.