Visual Servoing for Flexible Robot Manipulators

Project Report

Daqian Cheng; Haowen Shi; Hengrui Zhang

Abstract

In this project, we propose a method to solve IK problem for robot manipulators with flexible links. We formulate the problem as a IK for robot manipulators with unknown geometry, and iteratively and numerically estimate the twists of our manipulator. Our pipeline consists of two major parts: an AprilTag based end-effector pose estimation and iterative IK solver. We also show some experiment results at the end. We make our source available to the public[1].

1 Introduction

Robot manipulators are ubiquitous in the field of engineering, such as in manufacturing facilities like automobile factories or on the international space station assembling space telescopes. Whether the purpose is to find aliens or grab a soda in the vending machine, the nature of robot manipulator's task is to drive the end-effector to the desired position and orientation. However, it is not trivial to find for the desired pose a set of joint motion parameters. In the literature of robot kinematics, the problem of mapping from task space to joint space is recognized as the Inverse Kinematics (IK) problem, and there exist numerous methods, both analytical and numerical [1, 2], to address such a problem.

In this project, we propose a method to solve IK problem for robot manipulators with flexible links. Each link consists of a series of passive revolute friction joints, which are small and lightweight. They do not have motors and are not directly controlled, but are able to respond to abrupt external forces, such as collision with the environment. The purpose of having flexible links is to enable the robot manipulator to automatically adapt to the geometry of the environment. The intended application of such a robot manipulator is for confined space, where conventional robot manipulator may not be able to reach the goal pose because of collision with the environment. However, with the proposed flexible-link, the robot is able to adapt its shape to the geometry of the environment every time it comes in contact with the environment. Plus, the flexible links are able to absorb impact so that there will be a less direct impact on joint motors. There also exist other purposes in the literature for having different kinds of flexible links, for instance increasing safety for human-robot-interaction [3].

To solve the IK problem for the proposed robot manipulator, we propose to iteratively and numerically estimate the twists of our manipulator. When solving IK problems, it is generally assumed that the parameters of the manipulator links are given, usually in the form of twists. Twists directly come from the geometric configuration of the robot and remain constant values. However, if the links of manipulators are non-rigid, then the twists will also change. To overcome this issue, we numerically estimate the twists and constantly check if the current twists are valid using forward kinematics (FK). The tool frame pose is estimated with visual measurements of a AprilTag [4] attached to the environment. We demonstrate and evaluate our proposed robot and algorithms with simulated experiments.

[1] https://github.com/harveybia/jacobian-visual-servo

2 Model Selection

2.1 Problem Formulation

We formulate the problem as IK for robot manipulators with unknown geometry, such that given a goal pose for the end-effector $\mathbf{g}_d$, the robot will iteratively reconfigure itself to reach $\mathbf{g}_d$ in spite link geometry changes. We make the realistic assumptions that the number of joints is known and the transformation from world frame attached to the AprilTag to the manipulator's base frame $\mathbf{g}_{sw}$ and the hand-eye transformation (from camera frame to tool frame) $\mathbf{g}_{tc}$ are both known and fixed. We also assume that the flexible links are consisted of series of passive friction revolute joints with enough friction such that the links will remain rigid under normal manipulator movement and gravity, but will change shape according to large impacts with the environment. We operate under the assumption that we know the number of motorized joints as well as being able to read their joint angles and control them.

2.2 Approach

2.2.1 Framework

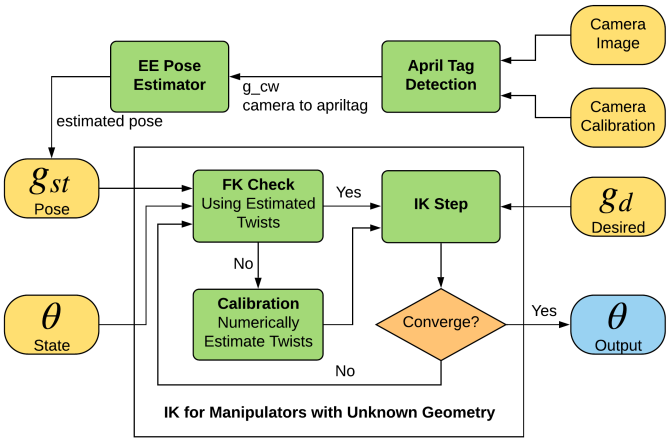

Figure 1: Framework of IK system.

The framework of our pipeline is shown in Fig. 1, which consists of two major parts: end-effector pose estimation and

iterative IK solver.

After receiving $\hat{\mathbf{g}}_{st}^{\mathrm{C}}$ estimation from camera, the IK solver first checks the validity of current twist estimation $\hat{\boldsymbol{\xi}}$. If $\hat{\mathbf{g}}_{st}^{\mathrm{C}}$ matches the $\hat{\mathbf{g}}_{st}^{\mathrm{FK}}(\hat{\boldsymbol{\xi}}, \boldsymbol{\theta})$ obtained by FK, then the current twist estimation is accurate and the IK solver carries out one step of $\boldsymbol{\theta}$ update, which will gradually drive the robot to $\mathbf{g}_d$. If the twists are invalid, then the IK solver performs a numerical twists estimation before solving the IK step. After executing the IK step, IK solver checks if the target pose $\mathbf{g}_d$ is reached. If yes, the system finishes and outputs the resulting joint angles.

2.2.2 End-Effector Pose Estimation

In this visual servoing pipeline we proposed to use an AprilTag to estimate the end-effector pose.

We assign the following frames: frame s is base link; frame t is end-effector; frame c is camera link; frame w is AprilTag.

Assuming that we know the static transforms between robot base link and AprilTag link ($\mathbf{g}_{sw}$), as well as the transform between end-effector link and camera link ($\mathbf{g}_{tc}$), the AprilTag will provide the transform between the camera frame and AprilTag frame ($\mathbf{g}_{cw}$). From this, we can have the target end-effector pose w.r.t. base link using the following equation:

$$\mathbf{g}_{st} = \mathbf{g}_{sw}\mathbf{g}_{cw}^{-1}\mathbf{g}_{tc}^{-1} \tag{1}$$

After some experimentation, we found out that the AprilTag reading is noisy, as the target pose reading jumps frequently, causing the algorithm to re-evaluate local Jacobian. Thus, we decided to use the simulation ground truth end-effector pose to work aournd this issue temporarily.

2.2.3 Manipulator Twist Estimation

In order to estimate the twists of the manipulator, we first numerically estimate the spatial Jacobian matrix $\mathbf{J}_{st}^{S}$ and then recover the twists from $\mathbf{J}_{st}^{S}$.

The Jacobian matrix maps motion in joint space to motion in task space, as described in Eq. 2.

$$\mathbf{V}_{st}^{S} = \left(\dot{\mathbf{g}}_{st}\mathbf{g}_{st}^{-1}\right)^{\vee} = \mathbf{J}_{st}^{S}\dot{\boldsymbol{\theta}} \tag{2}$$

We numerically estimate the Jacobian matrix $\mathbf{J}_{st}^{S}$ with small incremental motion $d\theta$ at each joint. From Eq. 2 we know that the ith column of $\mathbf{J}_{st}^{S}$ corresponds to the motion of the ith joint. Therefore, we are able to derive the discrete result of Jacobian estimation in Eq. 3.

$$\mathbf{J}_{st}^{S} = \begin{bmatrix} \frac{\left[(\mathbf{g}_{st}^{\mathrm{C}}(\boldsymbol{\theta}_0) - \mathbf{g}_{st}^{\mathrm{C}}(\boldsymbol{\theta}))\mathbf{g}_{st}^{\mathrm{C}}(\boldsymbol{\theta})^{-1}\right]^{\vee}}{d\theta} \\ \cdots \\ \frac{\left[(\mathbf{g}_{st}^{\mathrm{C}}(\boldsymbol{\theta}_n) - \mathbf{g}_{st}^{\mathrm{C}}(\boldsymbol{\theta}))\mathbf{g}_{st}^{\mathrm{C}}(\boldsymbol{\theta})^{-1}\right]^{\vee}}{d\theta} \end{bmatrix}^{\top} \tag{3}$$

In Eq. 3, $\boldsymbol{\theta}_i$ is the joint angles with a small motion $d\theta$ added to the ith joint. The accuracy of the numerical estimation of $\mathbf{J}_{st}^{S}$ is dependent on the step size $d\theta$, since small step may be vulnerable to noise while large step change will violate the linear approximation assumption. In our simulated experiment, we set $d\theta = 0.001$ rad.

We then estimate the twists $\hat{\boldsymbol{\xi}}$ from $\mathbf{J}_{st}^{S}$. Normally, we can compute the Jacobian directly from the twists using Eq. 4.

$$\mathbf{J}_{st}^{S}(\boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{\xi}_1 & \boldsymbol{\xi}_2' & \cdots & \boldsymbol{\xi}_n' \end{bmatrix}, \quad \boldsymbol{\xi}_i' = \mathrm{Ad}_{(e^{\hat{\boldsymbol{\xi}}_1\theta_1}\cdots e^{\hat{\boldsymbol{\xi}}_{i-1}\theta_{i-1}})}\boldsymbol{\xi}_i \tag{4}$$

Since $\boldsymbol{\theta}$ is assumed to be known to the system, we can recover $\hat{\boldsymbol{\xi}}$ from Jacobian $\mathbf{J}_{st}^{S}$ by deriving Eq. 5.

$$\begin{aligned} \hat{\boldsymbol{\xi}}(\mathbf{J}) &= \begin{bmatrix} \hat{\boldsymbol{\xi}}_1 & \hat{\boldsymbol{\xi}}_2 & \cdots & \hat{\boldsymbol{\xi}}_n \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{J}_1 & \mathrm{Ad}^{-1}_{(e^{\hat{\boldsymbol{\xi}}_1\theta_1})}\mathbf{J}_2 & \cdots & \mathrm{Ad}^{-1}_{(e^{\hat{\boldsymbol{\xi}}_1\theta_1}\cdots e^{\hat{\boldsymbol{\xi}}_{n-1}\theta_{n-1}})}\mathbf{J}_n \end{bmatrix} \end{aligned} \tag{5}$$

Where $\mathbf{J}_i$ is the ith column of the matrix $\mathbf{J}$.

The manipulator twist estimation process is triggered when $\hat{\mathbf{g}}_{st}$ from FK does not match that from visual estimation, which typically occurs when the robot links deformed from impact. At this point, the reference $\mathbf{g}_{st}^{0}$ is no longer $\mathbf{g}_{st}(\boldsymbol{\theta} = 0)$, therefore we reset the reference value of $\mathbf{g}_{st}$ as well as $\boldsymbol{\theta}$, namely $\mathbf{g}_{st}^{0}$ and $\boldsymbol{\theta}^{0}$, to their current values. The reference value $\mathbf{g}_{st}^{0}$ are used in FK process, and the reference value $\boldsymbol{\theta}^{0}$ is subtracted from incoming joint angle data for reference consistency.

2.2.4 Twist Accuracy Evaluation

We evaluate if the current twists estimation $\hat{\boldsymbol{\xi}}$ is accurate by performing forward kinematics (FK) using $\hat{\boldsymbol{\xi}}$ and joint angles $\boldsymbol{\theta}$ to obtain the transformation estimation from tool frame to base frame. The FK calculation is carried out with Eq. 6.

$$\hat{\mathbf{g}}_{st}^{\mathrm{FK}}(\hat{\boldsymbol{\xi}}, \boldsymbol{\theta}) = \prod_{i=1}^{n} e^{\hat{\boldsymbol{\xi}}_i^{\wedge}\theta_i}\mathbf{g}_{st}^{0} \tag{6}$$

In Eq. 6, $\hat{\boldsymbol{\xi}}_i^{\wedge}$ is the matrix representation of the twist of ith joint; $\boldsymbol{\theta}_i$ is the joint angle of the ith joint; $\mathbf{g}_{st}^{0}$ is the initial $\mathbf{g}_{st}$ associated with the current twist estimation and zero joint angles, which is defined and computed in the numerical twist estimation process, as described in Sec. 2.2.3.

After calculating the FK, we check if the transformation obtained from tool frame to base frame $\hat{\mathbf{g}}_{st}^{\mathrm{FK}}(\hat{\boldsymbol{\xi}}, \boldsymbol{\theta})$ matches that estimated by computer vision $\hat{\mathbf{g}}_{st}^{\mathrm{C}}$. With SE(3) composed with rotation and translation $g = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ 0 & 1 \end{bmatrix}$, the error between the two SE(3) is defined in Eq. 7.

$$\mathbf{e}(\mathbf{g}_1, \mathbf{g}_2) = \begin{bmatrix} e_R(\mathbf{R}_1, \mathbf{R}_2) \\ e_p(\mathbf{p}_1, \mathbf{p}_2) \end{bmatrix} = \begin{bmatrix} \arccos(\frac{\mathrm{tr}(\mathbf{R}_1\mathbf{R}_2^\top)-1}{2}) \\ \|\mathbf{p}_1 - \mathbf{p}_2\| \end{bmatrix} \tag{7}$$

If $\mathbf{e}\left(\hat{\mathbf{g}}_{st}^{\mathrm{FK}}(\hat{\boldsymbol{\xi}}, \boldsymbol{\theta}), \hat{\mathbf{g}}_{st}^{\mathrm{C}}\right)$ is smaller than a threshold value, the twists $\hat{\boldsymbol{\xi}}$ are considered to be accurate; if not, we perform the numerical estimation of twists.

2.2.5 Inverse Kinematics

The IK problem is solved numerically and iteratively as described in [1]. Suppose the desired pose $\mathbf{g}_d$ is decomposed into rotation quaternion $\mathbf{Q}_d$ and translation $\mathbf{p}_d$, at each step, an incremental step of the joint angles $\mathbf{d}\boldsymbol{\theta}$ is computed using Eq. 8.

$$\begin{cases} \mathbf{d}\boldsymbol{\theta} = \mathbf{k}\dot{\boldsymbol{\theta}} = \mathbf{J}_{st}^S(\boldsymbol{\theta})^{-1} \begin{bmatrix} \mathbf{k}_p\mathbf{e}_p \\ \mathbf{k}_o\mathbf{e}_o \end{bmatrix} \\ \mathbf{e}_p = \mathbf{p}_d - \mathbf{p}_{st} \\ \mathbf{Q} = \mathbf{Q}_d \times \mathbf{Q}_{st}^{-1} = (q_0, \mathbf{q}) \\ \mathbf{e}_o = q_0\mathbf{q} \end{cases} \tag{8}$$

In Eq. 8, $\mathbf{k}_p$ and $\mathbf{k}_o$ are gain matrices which determine the gain of IK. The inverse of $\mathbf{J}_{st}^S$ is computed using pseudo-inverse to avoid singularity.

3 Result

3.1 Experiment Setup

We implemented a simulation scene as shown in Fig. 2 in CoppeliaSim [5] (also known as V-REP) to benchmark our pipeline. In the simulated scene we have a 9-DoF snake robot which we mounted onto a platform and used as a robot manipulator.

To simulate the flexible links, we added a passive friction joint on top of each active motor joints. The active motor joints are controlled using CoppeliaSim's built-in position PID control while the passive joints have constant static friction (torque) and are not driven. This way, when the arm receives enough external force to overcome the static friction, the passive friction joints would turn, changing the arm's kinematic parameters. During the experiment, we make sure not to apply too much torque when driving the arm to a desired state (by servoing slowly) so that the friction joints do not turn.

We use Vortex Studio's [6] physics and dynamics engine to simulate joint friction and the arm dynamics.

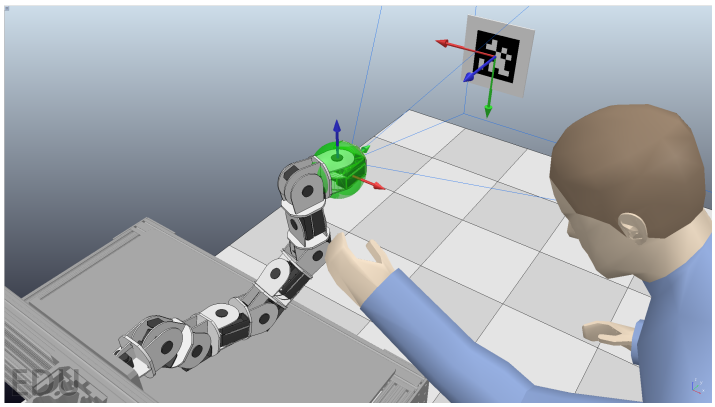

Figure 2: Simulation scene

3.2 Experiment Result

We evaluate the effectiveness of our algorithm by checking how well it converges on different target poses. Due to limited pages of this report we have posted our experiment results as a video here: https://youtu.be/B46CvOQX2VM

Our algorithm converges well on target poses with no rotational difference than current pose. With a rotational difference the convergence will take a longer time.

4 Discussion

In this work we demonstrated solving the Iterative Kinematics problem for robot manipulators with unknown geometry. Similar efforts have been made in the field. [7] addressed the problem of uncertain Jacobian matrix by introducing an uncertainty term into the Jacobian matrix. [8] employed the same approach and, like in this report, also use a camera to determine end-effector's pose. However, [8] used a feature-based computer vision approach instead of the marker-based approach used in this report. The feature-based approach does not require any marker from the environment and will thus be easier for generic applications, however with the problem of inferior accuracy and accumulative drift in pose estimation.

Here are some future work to further improve our pipeline:

1. More stable pose estimation for manipulator end-effector. Our current AprilTag approach does not provide robust and stable pose estimation.
2. Use a smarter and faster way to numerically estimate Jacobian, since currently this estimation process consumes a considerable amount of time.
3. Constant optimization of twist parameters, since pose $\mathbf{g}_{st}$ and joint angles $\boldsymbol{\theta}$ convey partial information of the twists $\boldsymbol{\xi}$.

References

[1] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[2] Samuel R Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17(1-19):16, 2004.

[3] Arthur Lismonde, Valentin Sonneville, and Olivier Brüls. Trajectory planning of soft link robots with improved intrinsic safety. *IFAC-PapersOnLine*, 50(1):6016–6021, 2017.

[4] John Wang and Edwin Olson. Apriltag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198. IEEE, 2016.

[5] E. Rohmer, S. P. N. Singh, and M. Freese. Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013. www.coppeliarobotics.com.

[6] Vortex studio: Real-time simulation software, Mar 2020.

[7] Chien Chern Cheah, Sadao Kawamura, and Suguru Arimoto. Feedback control for robotic manipulator with an uncertain jacobian matrix. *Journal of Robotic Systems*, 16(2):119–134, 1999.

[8] Chin-Su Kim, Eun-Jong Mo, Sung-Min Han, Min-Seok Jie, and Kang-Woong Lee. Robust visual servo control of robot manipulators with uncertain dynamics and camera parameters. *International Journal of Control, Automation and Systems*, 8(2):308–313, 2010.