Position-based Visual Servoing with Dual Manipulators

Tri Bien [1]

[1]Electrical and Computer Engineering, Vietnamese German University

Abstract

The research focuses on designing a system for position-based visual servoing (PBVS) with dual 6-Degree of Freedom (DoF) UR10e robots. The main robot uses computer vision integrated position-based servoing algorithm for tracking the ARTag trajectories from another robot by using Realsense D435 camera. The PBVS algorithm allows the robot to compute the inverse Jacobian matrix from the elementary transform sequence (ETS) UR10e robot model and spatial velocity from the ARTag pose. The result of this algorithm is joint velocity control on the real UR10e robot, which helps the main robot simultaneously track the ARTag trajectories in real time. The resulting video can be accessed at: triknight.github.io

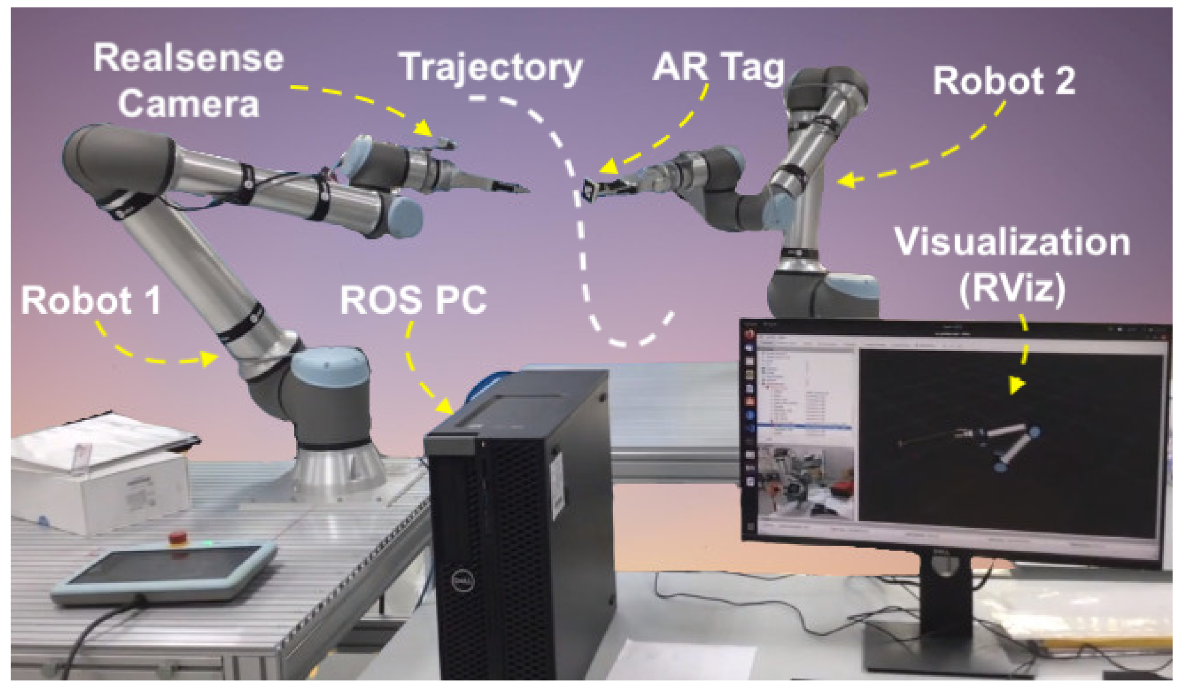

Figure 1. Hardware setup of position-based visual servoing with dual manipulators

The contributions of this paper are:

- Developed forward kinematics of the 6-DoF UR10e robot based on elementary transform sequence (ETS). The ETS model gets better robot kinematic representation, which can avoid the frame assignment constraints of Denavit and Hartenberg (DH) notation.
- Designed a position-based visual servoing (PBVS) with dual manipulators system by using ROS middle-ware, interfacing joint position, ARTag position, and controlling joints velocity of the real robot. Visualization of real robot operation data feedback by Rviz, and using Robotics Toolbox for simulation.
- Applied resolved-rate motion control (RRMC) into position-based servoing by calculating the inverse Jacobian matrix and spatial velocity from an error position vector of the end-effector.

System Overview

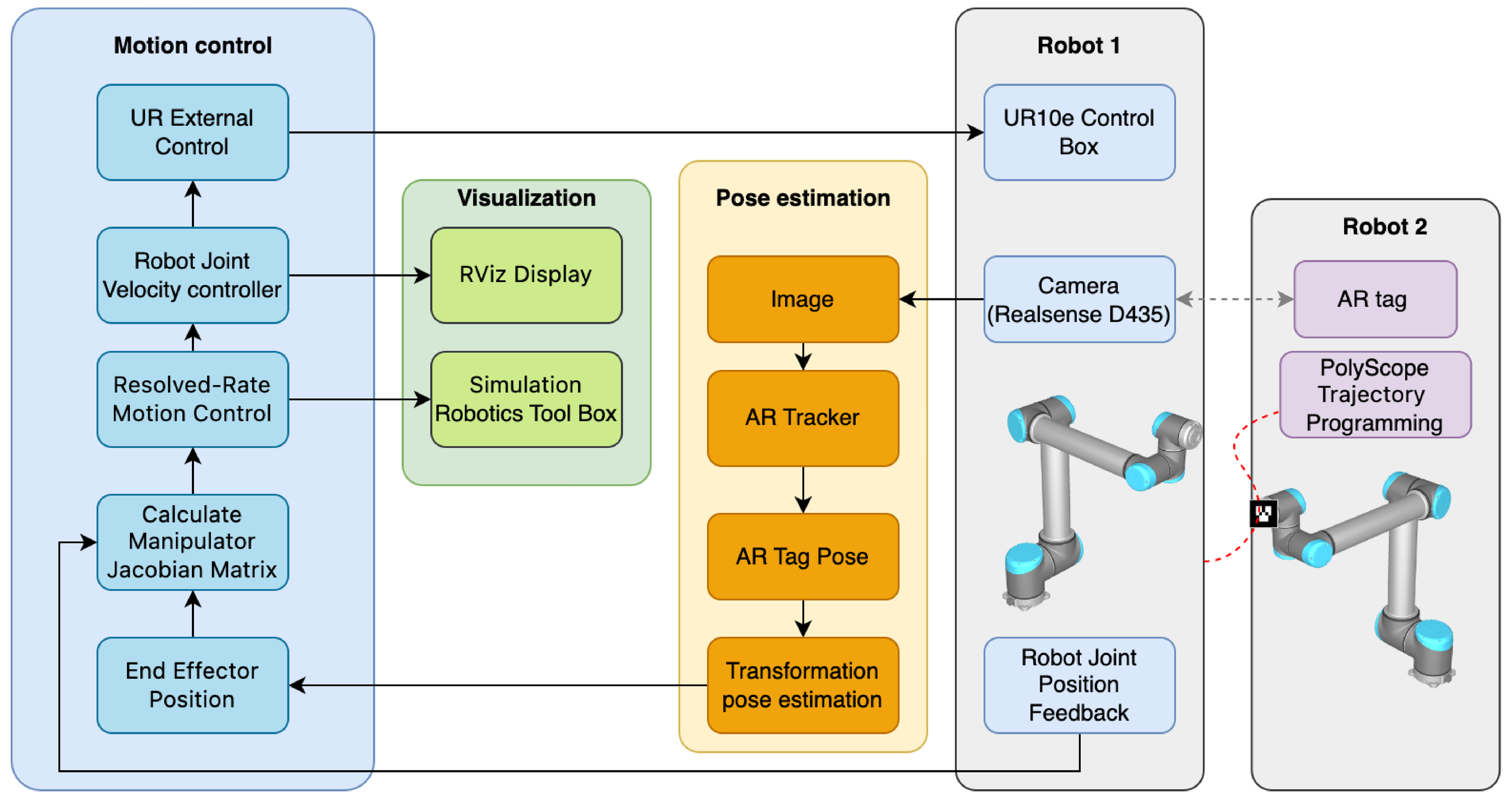

Figure 2. Position-based visual servoing with dual manipulators system

System Overview (Cont.)

The scheme of the developed PBVS with dual manipulators system is shown in Fig.2, which includes two robot arms. Robot 2 has been programming for grasping the ARTag and creating trajectory movement. Robot 1 has been connecting with a ROS computer with implemented PBVS algorithm. This algorithm can separate into three main components, which are pose estimation, motion control, and visualization.

1. Pose estimation: A fiducial marker -ARtag has been chosen to simplify object tracking and pose estimation. A new URDF model of the UR10e robot is created by integrating camera Realsense and gripper On-Robot RG2, which model can help the robot estimate the position from ARtag to the robot's end-effector via the open-source ARTag tracking ROS package [4].
2. Motion control: The position-based servoing algorithm includes two main steps. In the first step, the algorithm would be designed so that it would calculate the end-effector spatial velocity by running in as many iteration loops as necessary until the solution converges to the minimizing translation and rotation error vector between the end-effector current pose and ARTag pose. The second step, using resolved rate motion control calculates the required joint velocities to achieve the desired end-effector velocity by using an inverse Jacobian matrix with joint state feedback.
3. Visualization: The development of the visualization software is mostly done in ROS [5] using RViz to visualize the robot, simulation in Python Robotics Toolbox [1] and IDE environment in Visual Studio using several open source libraries (OpenCV, ARTag tracking).

Elementary Transform Sequence (ETS)

The elementary transform sequence (ETS) of the 6-DoF UR10e manipulator in its home configurations. E_i represents an elementary transform and ${}^{n}T_m$ represents the pose of link frame m in the reference frame of link n.

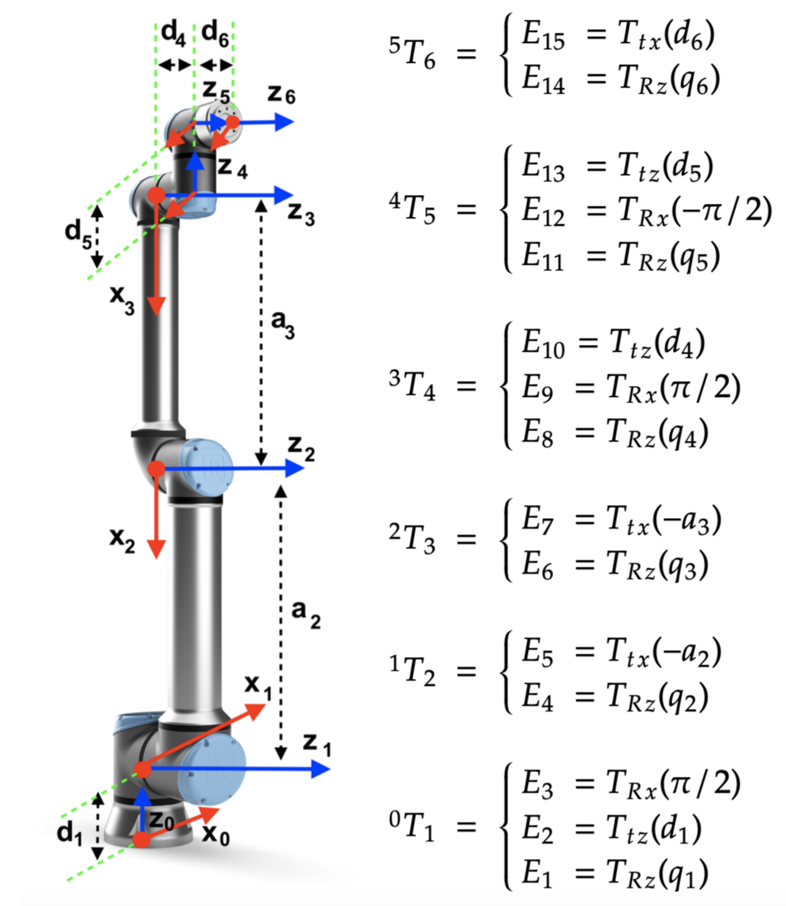

Figure 3. The Elementary Transform Sequence of the 6 DoF of UR10e robot

The dimension of the UR10e robot follows the table below. Where a_i, d_i are dimensions from joint i to join $i-1$, and q_i is the rotation around z_i axis of the robot joint.

Joint	a(m)	d(m)	α(rad)
Joint 1	0	0.1807	$\frac{\pi}{2}$
Joint 2	0.6127	0	0
Joint 3	0.57155	0	0
Joint 4	0	0.17415	$\frac{\pi}{2}$
Joint 5	0	0.11985	$-\frac{\pi}{2}$
Joint 6	0	0.11655	0

Table 1. These parameters reference the Universal Robot Denavit–Hartenberg parameters [6]

Position-Based Servoing

The elementary transform sequence (ETS) is another representation of the kinematics for a serial-link manipulator, which was introduced by Corke [2]. The forward kinematics of a manipulator presents a non-linear mapping between the robot's joint space and Cartesian task space.

$$ {}^{0}\boldsymbol{T}_e(t) = \mathcal{K}(\boldsymbol{q}(t)) \tag{1}$$

Where ${}^{0}\boldsymbol{T}_e \in \boldsymbol{SE(3)}$ is a homogeneous transformation matrix describing the pose of end-effector in the world-coordinate frame, $\mathcal{K}(.)$ is the product of a number of elementary transforms, and $\boldsymbol{q}(t) \in \mathbb{R}^n$ is the vector of joint generalized coordinates. Fig.3 is shown the ETS model of the UR10e and all dimension parameters as shown in Table.1.

Resolved-Rate Motion Control (RRMC)

End-effector spatial velocity in the world frame $\nu = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$ can be computed

$$\nu = \begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} J_v(q) \\ J_\omega(q) \end{pmatrix} \dot{q} = J\dot{q} \tag{2}$$

Where $J_v(q) \in \mathbb{R}^{3\times n}$, $J_v(\omega) \in \mathbb{R}^{3\times n}$ are the translational part and rotational part of the manipulator Jacobian matrix, and $\dot{q}$ is the joint velocities. RRMC is an elegant motion method along an arbitrarily oriented straight line in space [7], from Eq.2 can re-arrangement:

$$\dot{q} = J(q)^{-1}\nu \tag{3}$$

UR10e has 6 degree-of-freedom (DOF), so the Jacobian matrix $J(q) \in \mathbb{R}^{6\times 6}$ is square and invertible.

Position-based servoing (PBS)

PBS algorithms depend on an error vector that presents the translation and rotation from the end-effector's current pose to the ARTag pose [3].

$$\boldsymbol{e} = \begin{pmatrix} \tau({}^{0}T_{e}*) - \tau({}^{0}T_e) \\ \alpha(\rho({}^{0}T_{e}*) - \rho({}^{0}T_e)) \end{pmatrix} \in \mathbb{R}^6 \tag{4}$$

Where ${}^{0}T_e$ and ${}^{0}T_{e}*$ are forward kinematics and desired end-effector pose in the base frame of the robot, τ and ρ are translation and rotation of the end-effector pose, and $\alpha(.)$ transforms a rotation matrix to Euler vector. At each time step, the PBS scheme is constructed by taking the error term from Eq.4 to set end-effector spatial velocity ν

$$\nu = k\boldsymbol{e} \tag{5}$$

Where k is a gain, which is typically a diagonal matrix. From Eq.5 and Eq.3 , we can calculate the joint velocities based on the Jacobian matrix and spatial velocity, then apply them to the real robot.

Conclusion and future work

This study developed position-based visual servoing algorithms for existing dual commercial industrial robot manipulators and encapsulated them into a ROS package. Encouraging outcomes have been presented based on unique real-world dual manipulators experiments, demonstrating the ability of the position visual servo control scheme.

The further application of this work is developing visual approaches focused on semantic understanding, vision-enabled intervention, and SLAM algorithms for archiving fully autonomous manipulation in industrial applications. Further study can apply different types of manipulators, such as underwater manipulation for ROV and AUV systems.

References

[1] Peter Corke and Jesse Haviland. "Not your grandmother's toolbox – the Robotics Toolbox reinvented for Python". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2021). DOI: 10.1109/ICRA48506.2021.9561366.

[2] Peter I. Corke. "A Simple and Systematic Approach to Assigning Denavit–Hartenberg Parameters". In: *IEEE Transactions on Robotics* 23.3 (2007), pp. 590–594. DOI: 10.1109/TRO.2007.896765.

[3] Jesse Haviland and Peter Corke. "Manipulator Differential Kinematics Part I: Kinematics, Velocity, and Applications". In: (July 2022). DOI: 10.48550/arXiv.2207.01796.

[4] *Open source AR tag tracking library*. http://wiki.ros.org/ar_track_alvar.

[5] Morgan Quigley et al. "ROS: an open-source Robot Operating System". In: (May 2009).

[6] *Universal Robot DH parameters*. https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/.

[7] Daniel E. Whitney. "Resolved Motion Rate Control of Manipulators and Human Prostheses". In: *IEEE Transactions on Man-Machine Systems* 10.2 (1969), pp. 47–53. DOI: 10.1109/TMMS.1969.299896.