Article

Not peer-reviewed version

A Novel Feedforward Youla Parameterization Method for Avoiding Local Minima in Stereo Image Based Visual Servoing Control

Rongfei Li * and Farhad Assadian

Posted Date: 20 February 2025

doi: 10.20944/preprints202502.1603.v1

Keywords: PnP problem; Sterero camera system; image-based visual servoing; eye-in-hand configuration; feedforward and feedback control; accurate camera pose

 doi:10.20944/preprints202502.1603.v1

Article

A Novel Feedforward Youla Parameterization Method for Avoiding Local Minima in Stereo Image Based Visual Servoing Control

Rongfei Li * and Francis Assadian

University of California, Davis, USA

* Correspondence: rfli@ucdavis.edu

Abstract: In robot navigation and manipulation, accurately determining the camera's pose relative to the environment is crucial for effective task execution. In this paper, we systematically prove that this problem corresponds to the Perspective-3-Point (P3P) formulation, where exactly three known 3D points and their corresponding 2D image projections are used to estimate the pose of a stereo camera. In image-based visual servoing (IBVS) control, the system becomes overdetermined, as the 6 degrees of freedom (DoF) of the stereo camera must align with 9 observed 2D features in the scene. When more constraints are imposed than available DoFs, global stability cannot be guaranteed, as the camera may become trapped in a local minimum far from the desired configuration during servoing. To address this issue, we propose a novel control strategy for accurately positioning a calibrated stereo camera. Our approach integrates a feedforward controller with a Youla parameterization-based feedback controller, ensuring robust servoing performance. Through simulations, we demonstrate that our method effectively avoids local minima and enables the camera to reach the desired pose accurately and efficiently.

Keywords: PnP problem; Sterero camera system; image-based visual servoing; eye-in-hand configuration; feedforward and feedback control; accurate camera pose

1. Introduction

Determining the accurate pose of the camera is a fundamental problem in robot manipulations, as it provides the spatial transformation needed to map 3D world points to 2D image coordinates. The task involving camera pose estimation is essential for various applications, such as augmented reality [1], 3D reconstruction [2], SLAM [3], and autonomous navigation [4]. This becomes especially critical when robots operate in unstructured, fast-changing, and dynamic environments, performing tasks such as human-robot interaction, accident recognition and avoidance, and eye-in-hand visual servoing. In such scenarios, accurate camera pose estimation ensures that visual data is readily available for effective robotic control [5].

A classic approach to estimating the pose of a calibrated camera is solving the Perspective-n-Point (PnP) problem [6], which establishes a mathematical relationship between a set of n 3D points in the world and their corresponding 2D projections in an image. To uniquely determine the pose of a monocular camera in space, it is a Perspective-4-Point (P4P) problem, where exactly 4 known 3D points and their corresponding 2D image projections are used. Bujnak et al. [7] generalize four solutions for P3P problem while giving a single unique solution existed for P4P problem in a fully calibrated camera scenario. To increase accuracy, modern PnP approaches considers more than three 2D-3D correspondences. Among PnP solutions, EPnP (Efficient PnP) method finds the optimal estimation of pose from a linear system that expresses each reference point as a weighted sum of four virtual control points [8]. Another advanced approach, SQPnP (Sparse Quadratic PnP) formulates the problem as a sparse quadratic optimization, achieving enhanced accuracy by minimizing a sparse cost function [9].

 doi:10.20944/preprints202502.1603.v1

In recent years, many other methods have been developed to show improved accuracy than PnP based methods. For instance, Alkhatib et al. [10] utilize Structure from Motion (SfM) to estimate a camera's pose by extracting and matching key features across various images taken from different viewpoints to establish correspondences. Moreover, Wang et al. [11] introduce visual odometry into camera's pose estimations based on the movement between consecutive frames. In addition, recent advancements in deep learning have led to the development of models, such as Convolutional Neural Networks, specifically tailored for camera pose estimation [12–14]. However, these advanced methods often come with significant computational costs, requiring multiple images from different perspectives for accurate estimation. In contrast, PnP-based approaches offer a balance between accuracy and efficiency, as they can estimate camera pose from a single image, making them highly suitable for real-time applications such as navigation and scene understanding.

In image-based visual servoing (IBVS) [15], the primary goal is to control a robot's motion using visual feedback. Accurate real-time camera pose estimation is crucial for making informed control decisions, particularly in eye-in-hand (EIH) configurations [16,17], where a camera is mounted directly on a robot manipulator. In this setup, robot motion directly induces camera motion, making precise pose estimation essential. Due to its computational efficiency, PnP-based approaches remain widely applied in real-world IBVS tasks [18–20]. The PnP process begins by establishing correspondences between 3D feature points and their 2D projections in the camera image. The PnP algorithm then computes the camera pose from these correspondences, translating the geometric relationship into a format that the IBVS controller can use. By detecting spatial discrepancies between the current and desired camera poses, the robot can adjust its movements accordingly.

However, PnP-based IBVS presents challenges for visual control in robotics. One key issue is that IBVS often results in an overdetermined system, where the number of visual features exceeds the number of joint variables available for adjustment. For example, at least four 2D-3D correspondences are needed for a unique pose solution [6], but a camera's full six-degree-of-freedom (6-DOF) pose means that a 6-DOF robot may need to align itself with eight or more observed features. In traditional IBVS [15], the interaction matrix (or image Jacobian) defines the relationship between feature changes and joint velocities. When the system is overdetermined, this matrix contains more constraints than joint variables, leading to redundant information. Research [15] suggests that this redundancy may cause the camera to converge to local minima, failing to reach the desired pose. Although local asymptotic stability is always ensured in IBVS, global asymptotic stability cannot be guaranteed when the system is overdetermined.

Many studies have explored solutions to mitigate the local minimum problem in IBVS. One approach, proposed by Nicholas et al. [21], introduces a switched control method, where the system alternates between different controllers to escape local minima and avoid singularities in the image Jacobian. Another strategy, developed by Chaumette et al. [22], utilizes a 2-1/2-D visual servoing technique, which combines image-based and position-based features. This integration allows the camera to navigate around local minima during motion execution. Roque et al. [23] implement a model predictive control (MPC) approach, optimizing the quadrotor's trajectory to enhance robustness against local minima by predicting and adjusting control inputs in real time.

While these methods achieve significant improvements in most scenarios, they also introduce computational challenges compared to traditional IBVS. The switched control method requires different control strategies tailored to specific dynamics, increasing the complexity of the overall control architecture [24]. The 2-1/2-D visual servoing method demands real-time processing of both visual and positional data, which can impose significant computational loads and limit performance in dynamic environments [25]. MPC approaches introduce additional computational overhead by requiring complex optimization at every time step, making real-time implementation costly [26].

In this paper, we focus on the PnP framework for determining and controlling the pose of a stereo camera within an image-based visual servoing (IBVS) architecture. In traditional IBVS, depth information between objects and the image plane is crucial for developing the interaction matrix. However, with a monocular camera, depth can only be estimated or approximated using various

 doi:10.20944/preprints202502.1603.v1

algorithms [15], and inaccurate depth estimation may lead to system instability. In contrast, a stereo camera system can directly measure depth through disparity between two image planes, enhancing system stability.

A key novelty of this paper is providing a systematic proof that stereo camera pose determination in IBVS can be formulated as a P3P (Perspective-3-Point) problem, which, to the best of our knowledge, has not been explored in previous research. Since three corresponding points, totaling nine coordinates, are used to control the six DoFs camera pose, the IBVS control system for a stereo camera is overdetermined, leading to the potential issue of local minima during control maneuvers. While existing approaches can effectively address local minima, they often introduce excessive computational overhead, making them impractical for high-speed real-world applications.

To address this challenge, we propose a feedforward-feedback control architecture. The feedback component follows a cascaded control loop based on the traditional IBVS framework [15], where the inner loop handles robot joint rotation, and the outer loop generates joint angle targets based on visual data. One key improvement in this work is the incorporation of both kinematics and dynamics during the model development stage. Enhancing model fidelity in the control design improves pose estimation precision and enhances system stability, particularly for high-speed tasks. Both control loops are designed using Youla parameterization [27], a robust control technique that enhances resistance to external disturbances. The feedforward controller takes target joint configurations, which are associated with the desired camera pose as inputs, ensuring a fast system response while avoiding local minima traps. Simulation results presented in this paper demonstrate that the proposed control system effectively moves the stereo camera to its desired pose accurately and efficiently, making it well-suited for high-speed robotic applications.

2. System Configuration

An eye-in-hand robotic system has been developed to precisely control the pose of a stereo camera system, as illustrated in Figure 1. The robotic manipulator is equipped with six revolute joints, allowing unrestricted movement of the camera across six degrees of freedom (DoFs)—three for positioning and three for orientation. Assume a set of fiducial markers is placed in the workspace, with their coordinates fixed and predefined in an inertial frame. Utilizing the Hough transform [28] in computer vision, these markers can be detected and localized by identifying their centers in images captured by the stereo camera system. The control system within the robotic manipulator aligns the camera to its desired pose by matching the detected 2D features in the current frame with target 2D features. Throughout this process, it is assumed that all fiducial markers remain within the camera's field of view. As depicted in Figure 1, multiple Cartesian coordinate systems are illustrated. The base frame {O} serves as an inertial reference fixed to the bottom of the robot manipulator, while the camera frame {C} is a body-fixed frame attached to the robot's end-effector.

 doi:10.20944/preprints202502.1603.v1

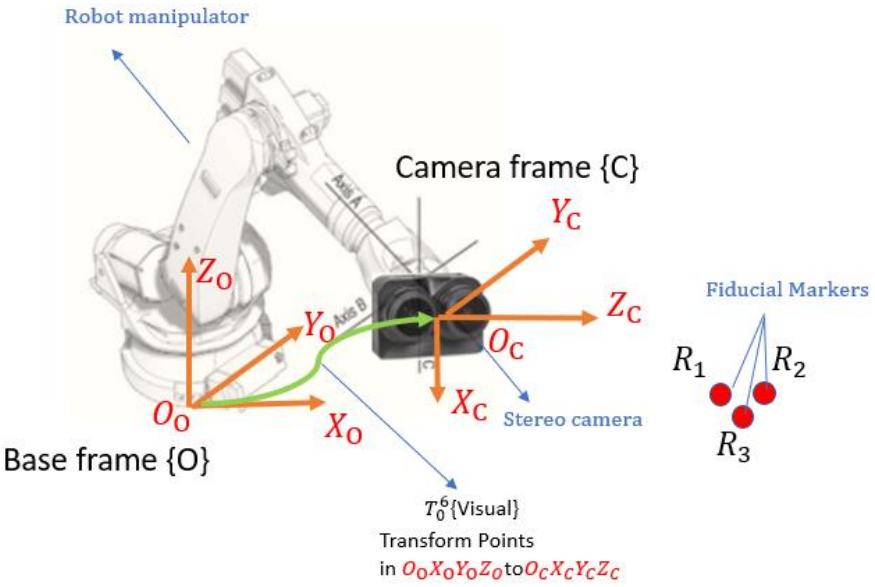


Figure 1. The eye-in-hand robot configuration.

3. Proof of P3P for the Stereo Camera System

Given its intrinsic parameters and a set of n correspondences between its 3D points and 2D projections to determine the camera's pose is known as perspective-n-point (PnP) problem. This well-known work [7] has proved that at least four correspondences are required to uniquely determine the pose of a monocular camera, a situation referred to as the P4P problem.

To illustrate, consider the P3P case for a monocular camera. Let points A, B, and C exist in space, with O_1, O_2 and O_3 representing different perspective centers. The angles ∠ AOB, ∠ AOC and ∠ BOC remain the same across all three perspectives. Given a fixed focal length, the image coordinates of points A, B, and C will be identical when observed from these three perspectives. In other words, it is impossible to uniquely identify the camera's pose based solely on the image coordinates of three points.

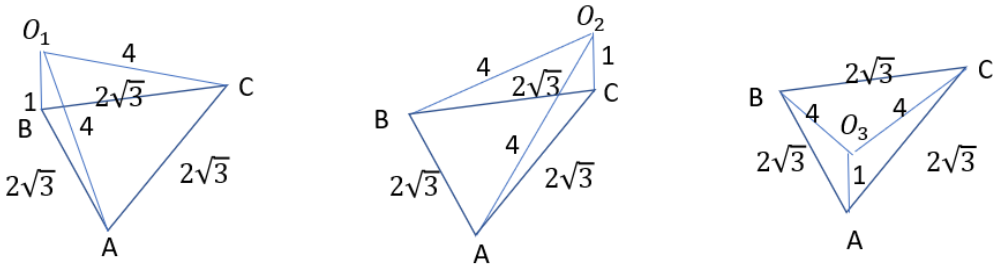


Figure 2. P3P case of a monocular camera.

The PnP problem with a stereo camera has not been thoroughly addressed in prior research. A stereo camera can detect three image coordinates of a 3D point in space. This paper proposes that a complete solution to the PnP problem for a stereo camera can be framed as a P3P problem. Below is the complete proof of this proposition.

Proof:

For a stereo system, if all intrinsic parameters are fixed and given, we can readily compute the 3D coordinates of an object point given the image coordinates of that point. This provides a unique mapping from the image coordinates of a point to its corresponding 3D coordinates in a Cartesian

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 February 2025 doi:10.20944/preprints202502.1603.v1

frame. The orientation and position of the camera system uniquely define the origin and axis orientations of this Cartesian coordinate system in space. Consequently, PnP problem can be framed as follows: given n points with their 3D coordinates measured in an unknown Cartesian coordinate system in space, what is the minimum number n required to accurately determine the position and orientation of the 3D Cartesian coordinate frame established in that space?

1) P1P problem with the stereo camera:

If we know the coordinates of a single point in space, defined by a 3D Cartesian coordinate system, an infinite number of corresponding coordinate systems can be established. Any such coordinate system can have its origin placed on the surface of a sphere centered at this point, with a radius $R = \sqrt{X^{C2} + Y^{C2} + Z^{C2}}$, where $[X^C, Y^C, Z^C]^T$ are the coordinates measured by the Cartesian system (see Figure 3).

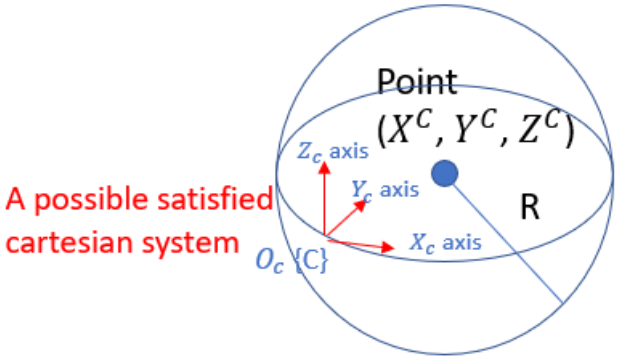
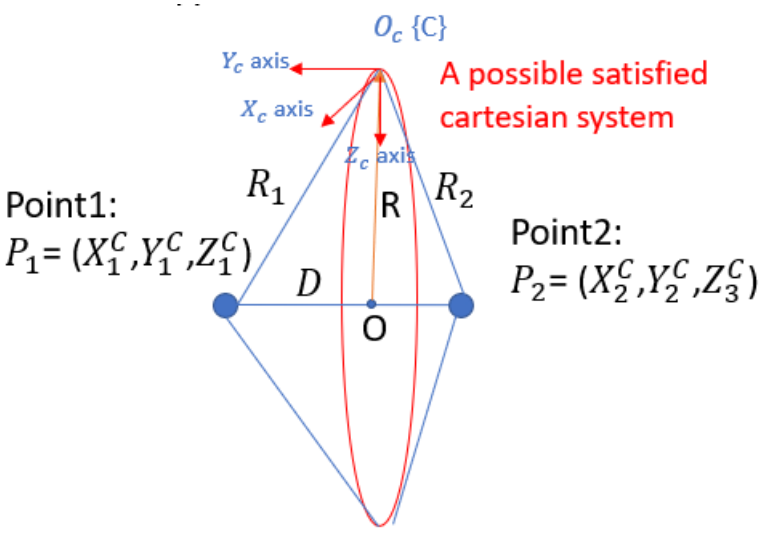


Figure 3. P1P Problem with a Stereo Camera System.

2) P2P problem with the stereo camera:

When the coordinates of two points in space are known, an infinite number of corresponding coordinate systems can be established. Any valid coordinate system can have its origin positioned on a circle centered at point O with a radius R as illustrated in Figure 4. This circle is constrained by the triangle formed by points O_C, P_1, and P_2, where the sides of the triangle are defined by the lengths R_1, R_2 and D. Specifically, $R_1 = \sqrt{X_1^{C2} + Y_1^{C2} + Z_1^{C2}}, R_2 = \sqrt{X_2^{C2} + Y_2^{C2} + Z_2^{C2}}$, $D = \sqrt{(X_1^C - X_2^C)^2 + (Y_1^C - Y_2^C)^2 + (Z_1^C - Z_2^C)^2}$.

The radius of the circle R corresponds to the height of the base D of the triangle. The center of the circle O is located at the intersection of the height and the base.



 doi:10.20944/preprints202502.1603.v1

Figure 4. P2P problem with a stereo camera system. All potential cartesian systems are located on the circle plotted in red.

3) P3P problem with the stereo camera:

When three points in space are known, and the lines connecting these points are not collinear, we can uniquely establish one coordinate system. As illustrated in the figure below, three non-collinear points define a plane in space, which has a uniquely defined normal unit vector $\vec{n}$. Given the coordinates of the three points, we can calculate vectors as follows: the vector $\overrightarrow{P_1P_2}$ = $(X_2^C - X_1^C, Y_2^C - Y_1^C, Z_2^C - Z_1^C)$, and the vector $\overrightarrow{P_1P_3}$ = $(X_3^C - X_1^C, Y_3^C - Y_1^C, Z_3^C - Z_1^C)$. The unit vector $\vec{n}$ which is perpendicular to the plane formed by these three points, can be expressed as:

$$\vec{n} = \frac{\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}}{|\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}|} \tag{1}$$

Here, × denotes the cross product.

The angles between $\vec{n}$ and XYZ axes of the coordinate system can be expressed as follows:

$$\cos(\theta_x) = \vec{n} \cdot \vec{\imath} \tag{2}$$

$$\cos(\theta_y) = \vec{n} \cdot \vec{\jmath} \tag{3}$$

$$\cos(\theta_z) = \vec{n} \cdot \vec{k} \tag{4}$$

Where θ_x, θ_y and θ_z are the angles between $\vec{n}$ and the unit vectors in the X, Y, and Z directions, denoted as $\vec{\imath}$, $\vec{\jmath}$, and $\vec{k}$ respectively. Therefore, with the direction $\vec{n}$ fixed in space, the orientations of each axis of the coordinate system can be computed uniquely.

According to the P1P problem, the origin of the coordinate system must lie on the surface of a sphere centered at P_1 with radius $= \sqrt{X_1^{C\,2} + Y_1^{C\,2} + Z_1^{C\,2}}$ as depicted in Figure 3. Each coordinate system established with a different origin point on the surface of this sphere results in a unique configuration of the axis orientations. Therefore, as the orientations of the axes are defined in space, the position of the frame (or the position of the origin) is also uniquely defined.

In conclusion, the P3P problem is sufficient to solve the PnP problem for a stereo camera system.

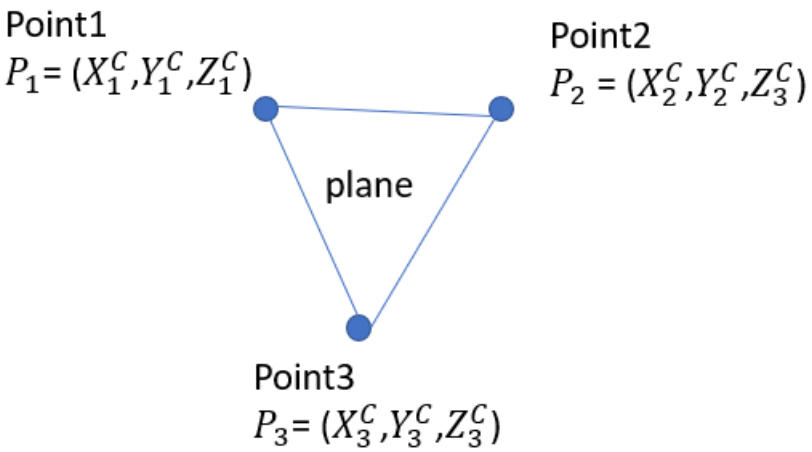


Figure 5. P3P Problem with a Stereo Camera System.

Prove Concluded

This proposition indicates that to uniquely determine the full 6 DoFs of the stereo camera, at least three points (or nine 2D features) are required to match in the image-based visual servoing control.

 doi:10.20944/preprints202502.1603.v1

4. Model Development

4.1. Stereo Camera Model

Depth between the objects to the camera plane is either approximated or estimated in the IBVS for generating the interaction matrix [15]. Using a stereo camera system in IBVS eliminates the inaccuracies associated with monocular depth estimation, as it directly measures depth by leveraging the disparity between the left and right images.

The stereo camera model is illustrated in Figure 6. A stereo camera consists of two lenses separated by a fixed baseline b. Each lens has a focal length F (measured in mm) which is the distance from the image plane to the focal point. Assuming the camera is calibrated, the intrinsic parameters: b, F is accurately estimated. A scene point I is measured in the 3D coordinate frame {C} centered at the middle of the baseline with its coordinates as $[X^C, Y^C, Z^C]^T$. The stereo camera model maps the 3D coordinates of this point to its 2D coordinates projected on the left and right image plane as $[u_l, v]^T$ and $[u_r, v]^T$, respectively. The full camera projection map, incorporating both intrinsic and extrinsic parameters, is given by:

$$s \cdot p_{image} = K \cdot [R|T] \cdot P_C \tag{5}$$

Where p_{image} are the image coordinates of the point and P_C are the 3D coordinates measured in the camera frame {C}. s is the scale factor that ensures correct projection between 2D and 3D features. K is the intrinsic matrix with a size of 3X3, and the mathematical expression is presented as:

$$K = \begin{bmatrix} F & k & u_0 \\ 0 & F & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Where k is the skew factor, which represents the angle between the image axes (u and v axis). u_0 and v_0 are coordinates offsets in image planes.

In Equation (5), R is the rotational matrix from camera frame {C} to each image coordinate frame, and T is the translation matrix from camera frame {C} to each camera lens center. Since there is no rotation between the camera frame {C} and image frames but only a translation along the X_C axis occurs, the transformation matrices for the left and right image planes are expressed as:

$$[R|T]_{Left} = \left[\begin{array}{ccc|c} 1 & 0 & 0 & -b/2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \tag{7}$$

$$[R|T]_{Right} = \left[\begin{array}{ccc|c} 1 & 0 & 0 & b/2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \tag{8}$$

Assume the u and v axis are perfectly perpendicular (take k = 0), and there are no offsets in the image coordinates (take $u_0 = v_0 = 0$) for both lens. Also, set factor $s = Z_I^C$ accounts for perspective depth scaling. The projection equations for the left and right image planes can be rewritten in homogeneous coordinates as:

$$Z_I^C \cdot \begin{bmatrix} u_l \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} F & 0 & 0 \\ 0 & F & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \left[\begin{array}{ccc|c} 1 & 0 & 0 & b/2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \cdot \begin{bmatrix} X_I^C \\ Y_I^C \\ Z_I^C \\ 1 \end{bmatrix} \tag{9}$$

$$Z_I^C \cdot \begin{bmatrix} u_r \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} F & 0 & 0 \\ 0 & F & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \left[\begin{array}{ccc|c} 1 & 0 & 0 & \text{-b/2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \cdot \begin{bmatrix} X_I^C \\ Y_I^C \\ Z_I^C \\ 1 \end{bmatrix} \tag{10}$$

Equations (9) and (10) establish the mathematical relationship between the 3D coordinates of a point in the camera frame {C} and its 2D projections on the left and right image planes. The pixel

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 February 2025 doi:10.20944/preprints202502.1603.v1

value along the v -axis remains the same for both images. As a result, a scene point's 3D coordinates can be mapped to a set of three image coordinates in the stereo camera system, expressed as:

$$\text{Stereo-camera mapping } M: \ P^C = [X^C, Y^C, Z^C]^T \rightarrow p_{image} = [u_l, u_r, v]^T \tag{11}$$

The mapping function M is nonlinear and depends on the stereo camera parameters P_{a_camera}, specifically b, and F.

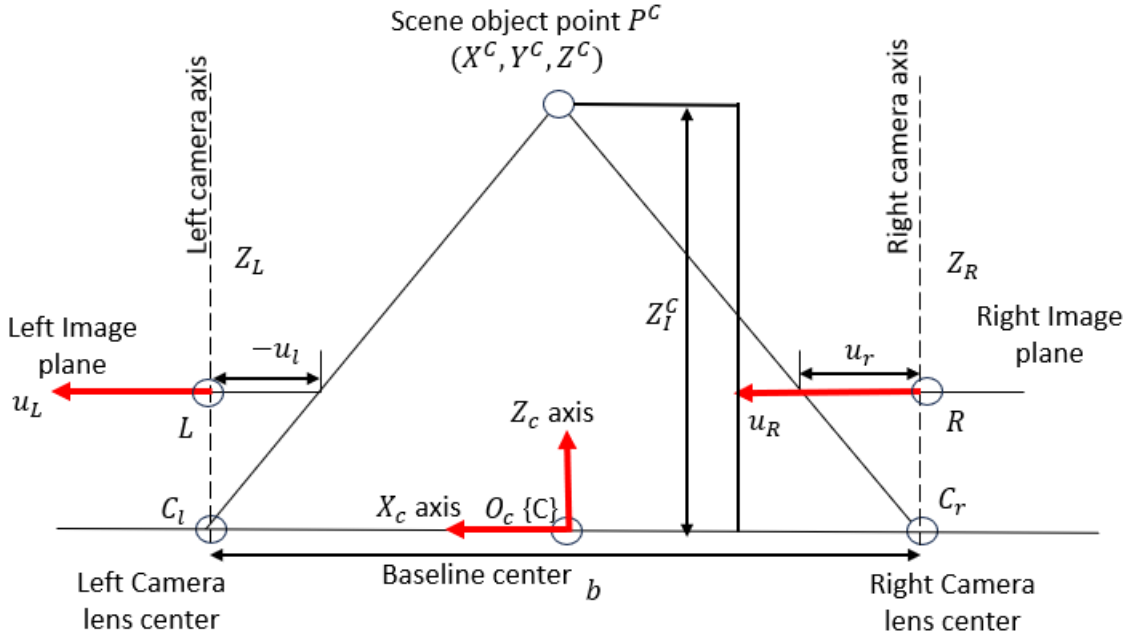


Figure 6. The projection of a scene object on the stereo camera's image planes. Note: The v-coordinate on each image plane is not displayed in this plot but is measured along the axis that is perpendicular to and pointing out of the plot.

4.2. Robot Manipulator Kinematic Model

A widely used method for defining and generating reference frames in robotic applications is the Denavit-Hartenberg (D-H) convention [29]. In this approach, each robotic link is associated with a Cartesian coordinate frame $O_iX_iY_iZ_i$. According to the D-H convention, the homogeneous transformation matrix A_i^{i-1}, which represents the transformation from frame $i-1$ to frame i, can be decomposed into a sequence of four fundamental transformations:

$$A_i^{i-1} = Rot_{z,q_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i}$$

Expanding the transformation into its matrix form:

$$A_i^{i-1} = \begin{bmatrix} c_{q_i} & -s_{q_i} & 0 & 0 \\ s_{q_i} & c_{q_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

$$= \begin{bmatrix} c_{q_i} & -s_{q_i}c_{\alpha_i} & s_{q_i}s_{\alpha_i} & a_i c_{q_i} \\ s_{q_i} & c_{q_i}c_{\alpha_i} & -c_{q_i}s_{\alpha_i} & a_i s_{q_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

 doi:10.20944/preprints202502.1603.v1

$$\text{Note: } c_{\theta_i} \equiv \cos(q_i),\ c_{\alpha_i} \equiv \cos(\alpha_i),\ s_{\theta_i} \equiv \sin(q_i),\ s_{\alpha_i} \equiv \sin(\alpha_i) \tag{14}$$

The parameters q_i, a_i, α_i and d_i define the link and joint characteristics of the robot. Here, a_i is the link length, q_i is the joint rotational angle, α_i is the twist angle, and d_i is the offset between consecutive links. The values for these parameters are determined following the procedure outlined in [29].

To compute the transformation from the end-effector frame $O_6X_6Y_6Z_6$ (denoted as {E}) to the base frame $O_0X_0Y_0Z_0$ (denoted as {O}), we multiply the individual transformations along the kinematic chain:

$$T_6^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 A_6^5 \tag{15}$$

Furthermore, the transformation matrix from the base frame {O} to the end-effector frame {E} can be derived by taking the inverse of T_6^0:

$$T_0^6 = (T_6^0)^{-1} \tag{16}$$

If a point P^O is defined in the base frame, its coordinates in the end-effector frame P^E can be found using:

$$P^E = T_0^6\, P^O \tag{17}$$

Assuming that the camera remains static relative to the end-effector, we introduce a constant transformation matrix T_E^C that maps points from the end-effector frame {E} to the camera frame {C}. For a stereo camera system, this camera frame is located at the center of the stereo baseline, as shown in Figure 6. The coordinates of a point in space, measured in the base frame, can then be expressed in the camera frame as:

$$P^C = T_E^C T_0^6\, P^O \tag{18}$$

Equation (18) describes how a given 3D point in the base frame $P^O = [X^O, Y^O, Z^O]$ is mapped to the camera frame $P^C = [X^C, Y^C, Z^C]$ using transformation:

$$\text{Robot manipulator mapping } \mathcal{H}:\ P^O = [X^O, Y^O, Z^O]^T \rightarrow P^C = [X^C, Y^C, Z^C]^T \tag{19}$$

The mapping function $\mathcal{H}$ is nonlinear and depends on the current joint angle of robot $q = [q_i | i \in 1,2,3,4,5,6]$, robot geometric parameter $P_{a\_robot} = [a_i,\ \alpha_i,\ d_i\ | i \in 1,2,3,4,5,6]$, and constant transformation matrix T_E^C.

4.3. Eye-in-Hand Kinematic Model

By combining the stereo camera mapping M from Equation (11) and the robot manipulator mapping $\mathcal{H}$ from Equation (19), we define a nonlinear transformation $\boldsymbol{\mathcal{F}}$, which maps any point measured in the base frame {O} to its image coordinates as captured by the stereo camera. This transformation is expressed as:

$$\text{Eye-in-hand mapping } \mathcal{F}:\ P^O = [X^O, Y^O, Z^O]^T \rightarrow p_{image} = [u_l, u_r, v]^T \tag{20}$$

The Mapping $\mathcal{F}$ is nonlinear and depends on variables current joint angles: q and parameters P_a, which includes stereo camera parameter P_{a_camera}, robot geometric parameter P_{a_robot}, and transformation matrix T_E^C. In other words:

$$\text{For any time } t \geq 0,\ p_{image} = \mathcal{F}(q(t), Pa, P^O) \tag{21}$$

$$Pa = [P_{a_{camera}}, P_{a_{robot}}, T_E^C] \tag{22}$$

 doi:10.20944/preprints202502.1603.v1

4.4. Robot Inverse Kinematic Model

Inverse kinematics determines the joint angles required to achieve a given camera pose relative to the inertial frame. The camera pose in the inertial frame can be expressed as a 4X4 matrix:

$$Pose^O = \begin{bmatrix} n_x & s_x & a_x & d_x \\ n_y & s_y & a_y & d_y \\ n_z & s_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{23}$$

Here, the vectors $[n_x, n_y, n_z]^T$, $[s_x, s_y, s_z]^T$ and $[a_x, a_y, a_z]^T$ represent the camera's directional vectors for Yaw, Pitch, and Roll, respectively, in the base frame {C}. Additionally, the vector $[d_x, d_y, d_z]^T$ denotes the absolute position of the camera center in the base frame {C}.

The camera pose in the camera frame {C} is straightforward as it can be expressed as another 4X4 matrix:

$$Pose^C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{24}$$

The nonlinear inverse kinematics problem involves solving for the joint angles q that satisfy the equation:

$$Pose^C = T_E^C \cdot T_0^6(q) \cdot Pose^O \tag{25}$$

where T_E^C is the transformation from the end-effector frame to the camera frame, and $T_0^6(q)$ represents the transformation from the base frame to the end-effector frame, which is a function of the joint angles q.

The formulas for computing each joint angle are derived from the geometric parameters of the robot. The results of the inverse kinematics calculations for the ABB IRB 4600 elbow manipulator [30], used for simulations in this paper, are summarized in Appendix A.

4.5. Robot Dynamic Model

Dynamic models are included in the inner joint control loop, which will be discussed in section 6. Without derivation, the dynamic model of a serial of 6-link rigid, non-redundant, fully actuated robot manipulator can be written as [31]:

$$(D(q) + J)\ddot{q} + (C(q, \dot{q}) + \frac{B}{r})\dot{q} + \mathrm{g}(q) = u \tag{26}$$

Where $q \in \mathbb{R}^{6X1}$ is the vector of joint positions, and $u \in \mathbb{R}^{6X1}$ is the vector of electrical power input from DC motors inside joints, $D(q) \in \mathbb{R}^{6X6}$ is the symmetric positive defined matrix, $C(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{6X6}$ is the vector of centripetal and Coriolis effects, $g(\boldsymbol{q}) \in \mathbb{R}^{6X1}$ is the vector of gravitational torques, $J \in \mathbb{R}^{6X6}$ is a diagonal matrix expressing the sum of actuator and gear inertias, $B \in \mathbb{R}^{6X1}$ is the damping factor, $r \in \mathbb{R}^{6X1}$ is the gear ratio.

5. Control Policy Diagram

Figure 7 illustrates the overall control system architecture, designed to guide the robot manipulator so that the camera reaches its desired pose, $\overline{pose^O} \in \mathbb{R}^{4X4}$, in the world space. To achieve this, three fiducial markers are placed within the camera's field of view, with their coordinates in the inertial frame pre-determined and represented as $P^O \in \mathbb{R}^{9X1}$. Using the robot's inverse kinematics and the eye-in-hand kinematic model, the expected image coordinates of these fiducial markers, when viewed from the desired camera pose, are computed as $\overline{p_{image}} \in \mathbb{R}^{9X1}$. These computed image coordinates serve as reference targets in the feedback control loop.

The IBVS framework is implemented within a cascaded feedback loop. In the outer control loop, the camera controller processes the visual feedback error, e_p, which represents the difference between the image coordinates of the fiducial points at the current and desired camera poses. Based

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 February 2025 doi:10.20944/preprints202502.1603.v1

on this error, the outer loop generates reference joint angles, $q_{feedback} \epsilon \mathbb{R}^{6X1}$, to correct the robot's configuration.

The inner control loop, shown in Figure 8, incorporates the robot's dynamic model to regulate the joint angles, ensuring they align with the commanded reference angles, $q_{ref} \epsilon \mathbb{R}^{6X1}$. However, due to the limitations of low-fidelity and inexpensive joint encoders, as well as inherent dynamic errors such as joint compliance, high frequency noises and low frequency model disturbances are introduced into the system. All sources of errors from the joint control loop are collectively modeled as an input disturbance, $d_{q_T} \epsilon \mathbb{R}^{6X1}$, which affects the outer control loop.

The feedforward control loop operates as an open-loop system, quickly bringing the camera as close as possible to its target pose, despite the presence of input disturbances. The feedforward controller outputs a reference joint angle command, $q_{feedforward} \epsilon \mathbb{R}^{6X1}$, which is sent to the inner loop to facilitate rapid convergence to the desired configuration.

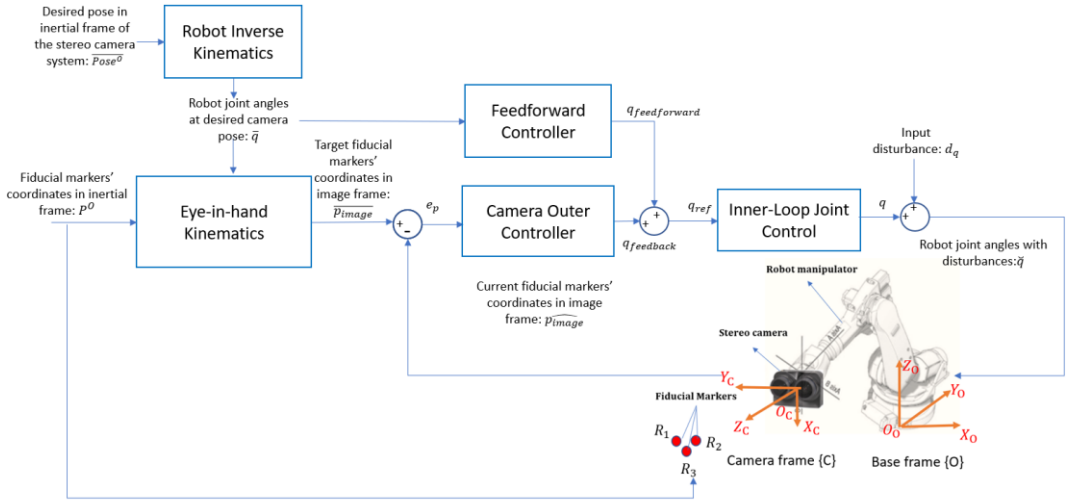


Figure 7. Feedforward-feedback control architecture. Note: In mathematics, the in-loop hardware is equivalent to the Eye-in-hand Kinematics Model.

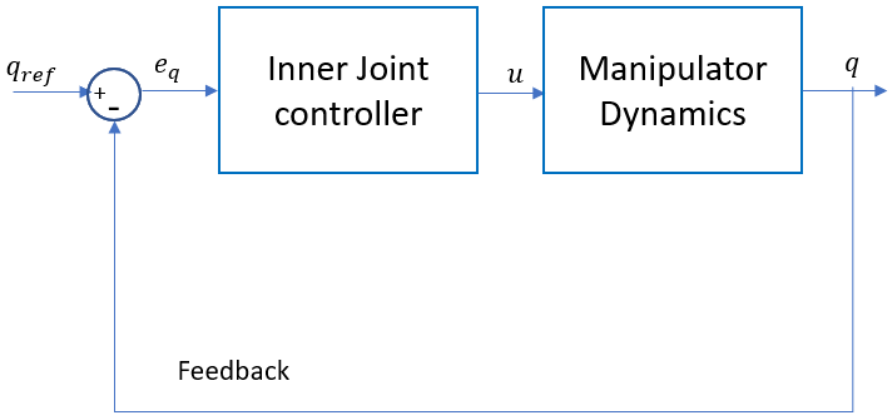


Figure 8. Inner joint angle control loop.

6. Controller Designs

6.1. Inner Joint Angle Control Loop

As shown in Figure 8, the primary goal of the inner joint controller is to stabilize the manipulator dynamics, which is expressed as a nonlinear Equation (26).

Simplify Equation (5.1) as follows:

 doi:10.20944/preprints202502.1603.v1

$$M(q)\ddot{q} + h(\mathrm{q}, \dot{\mathrm{q}}) = \mathrm{u} \tag{27}$$

With:

$$M(q) = D(q) + J \tag{28}$$

$$h(\mathrm{q}, \dot{\mathrm{q}}) = (C(q, \dot{q}) + \frac{B}{r})\dot{q} + \mathrm{g}(q) \tag{29}$$

Then, transform the control input as following:

$$u = M(q)v + h(\mathrm{q}, \dot{\mathrm{q}}) \tag{30}$$

where $v \in \mathbb{R}^{6X1}$ is a virtual input. Then, substitute for u in Equation (27) using Equation (30), and since $M(q) \in \mathbb{R}^{6X6}$ is invertible, we will have a reduced system equation as follows:

$$\ddot{q} = v \tag{31}$$

This transformation is feedback linearization technique with the new system equation given in Equation (31). This equation represents 6 uncoupled double integrators. The overall feedback linearization method is illustrated in Figure 9. In this control block diagram, the joint angle $q \in \mathbb{R}^{6X1}$ are forced to follow the target joint angle $q_R \in \mathbb{R}^{6X1}$. The Nonlinear interface transforms the linear virtual control input $v \in \mathbb{R}^{6X1}$ to the nonlinear control input $u \in \mathbb{R}^{6X1}$ by using Equation (30). The output of the manipulator dynamic model, the joint angles, $q \in \mathbb{R}^{6X1}$, and their first derivatives, $\dot{q} \in \mathbb{R}^{6X1}$, are utilized to calculate $M(q) \in \mathbb{R}^{6X6}$ and $h(\mathrm{q}, \dot{\mathrm{q}}) \in \mathbb{R}^{6X1}$ in the Nonlinear interface. The linear joint controller is designed using Youla parameterization technique [27] to control the nominally linear system in Equation (31).

The design of a linear Youla controller with nominally linear plant is presented next.

Since the transfer functions between all inputs to outputs in Equation (31) are the same and decoupled, it is valid to first design a Single Input and Single Output (SISO) controller and use the multiple of the same controller for a six-dimension to obtain the Multiple Input and Multiple Output (MIMO) version. In other words, first design a controller $G_{C_{SISO}}^{Inner}$ that satisfies:

$$v_{SISO} = \ddot{q}_{SISO} \tag{32}$$

where v_{SISO} is a single input to a nominally linear system and $\ddot{q}_{SISO}$ is the second order derivative of a joint angle. The controller in Figure 9 can be then written as:

$$G_{C_{MIMO}}^{Inner} = G_{C_{SISO}}^{Inner} \cdot I_{6X6} \tag{33}$$

where I_{6X6} is a 6×6 identity matrix. The transfer function of the SISO nominally linear system from Equation (31) is:

$$G_{p_{SISO}}^{Inner} = \frac{1}{s^2} \tag{34}$$

Note that $G_{p_{SISO}}^{Inner}$ has two Bounded Input Bounded Output (BIBO) unstable poles at origin. To ensure internal stability of the feedback loop, the closed loop transfer function, T_{SISO}, should meet the interpolation conditions [32]:

$$T_{SISO}^{inner}(s = 0) = 1 \tag{35}$$

$$\frac{dT_{SISO}^{inner}}{ds}|_{s=0} = 0 \tag{36}$$

Use the following relationship to compute a Youla transfer function: Y_{SISO} as:

$$T_{SISO}^{inner} = Y_{SISO}^{inner} G_{p_{SISO}}^{Inner} \tag{37}$$

The T_{SISO}^{inner} is designed so that it satisfies the conditions in Equations (35) and (36). The sensitivity transfer function, S_{SISO}^{inner}, is then calculated as follows:

$$S_{SISO}^{inner} = 1 - T_{SISO}^{inner} \tag{38}$$

Without providing the design details, the closed-loop transfer function can be in the following form to satisfy the interpolation conditions:

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 February 2025 doi:10.20944/preprints202502.1603.v1

$$T_{SISO}^{inner} = \frac{(3\tau_{in}s + 1)}{(\tau_{in}s + 1)^3} \tag{39}$$

Where τ_{in} specifies the pole and zero locations and represents the bandwidth of the control system. τ_{in} can be tuned so that the response can be fast with less-overshoot.

The next step is to derive $G_{C_{SISO}}^{Inner}$ from relationships between the closed-loop transfer function, T_{SISO}^{inner}, the sensitivity transfer function, S_{SISO}^{inner}, and the Youla transfer function, Y_{SISO}^{inner}, in Equations (40)–(42):

$$Y_{SISO}^{inner} = T_{SISO}^{inner} G_{pSISO}^{Inner\,-1} = \frac{s^2(3{\tau_{in}}^2 s + 1)}{(\tau_{in}s + 1)^3} \tag{40}$$

$$S_{SISO}^{inner} = 1 - T_{SISO}^{inner} = \frac{s^2({\tau_{in}}^3 s + 3{\tau_{in}}^2)}{(\tau_{in}s + 1)^3} \tag{41}$$

$$G_{C_{SISO}}^{Inner} = Y_{SISO}^{inner} S_{SISO}^{inner\,-1} = \frac{3{\tau_{in}}^2 s + 1}{{\tau_{in}}^3 s + 3{\tau_{in}}^2} \tag{42}$$

From Equation (33), a MIMO controller can be computed as follows:

$$G_{C_{MIMO}}^{Inner} = \frac{3{\tau_{in}}^2 s + 1}{{\tau_{in}}^3 s + 3{\tau_{in}}^2} \cdot I_{6\times6} \tag{43}$$

Equation (43) provides the expression of the inner joint controller and the closed loop of the inner loop can be expressed as:

$$T_{MIMO}^{inner} = \frac{(3\tau_{in}s + 1)}{(\tau_{in}s + 1)^3} \cdot I_{6\times6} \tag{44}$$

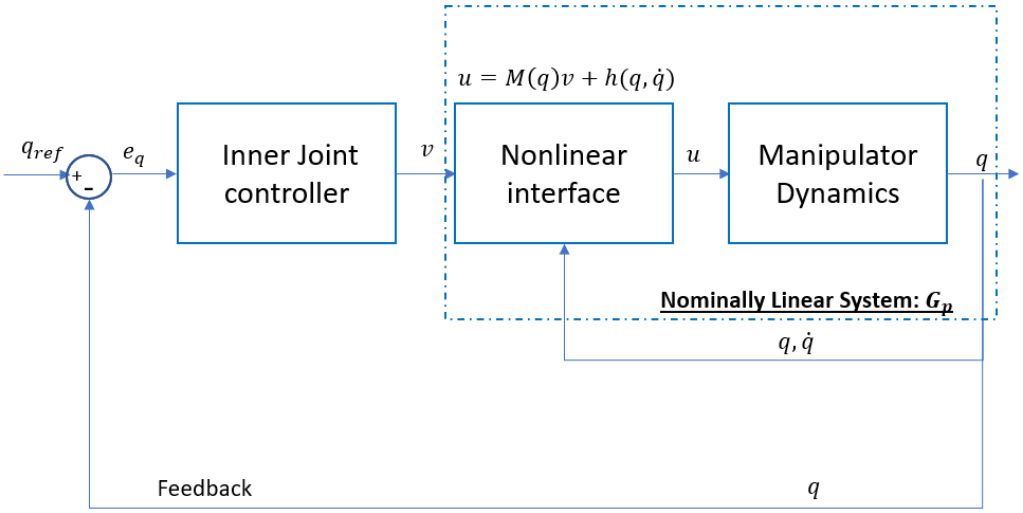


Figure 9. Feedback linearization Youla control design for inner loop.

6.2. Feedforward Control Loop

The feedforward loop is an open loop, disturbed by input disturbances as shown in Figure 10. Feedforward controller is the inverse process of Inner-loop Joint control loop T_{MIMO}^{inner}, whose closed-loop transfer function is given in Equation (44).

Therefore, the feedforward controller can be designed as

$$T_{forward} = \frac{1}{T_{inner-closed}} \frac{1}{(\tau_{forward}s+1)^2} = \frac{(\tau_{in}s+1)^3}{(3\tau_{in}s+1)} \frac{1}{(\tau_{forward}s+1)^2} \cdot I_{6\times6} \tag{45}$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 February 2025 doi:10.20944/preprints202502.1603.v1

The double poles s = -1/$\tau_{forward}$ are added to make $T_{forward}$ proper. Choose $\tau_{forward}$ so that the added double poles are 10 times larger than the bandwidth of the original improper $T_{forward}$. In other words, $\tau_{forward}$ is chosen as

$$\tau_{forward} = 0.1\tau_{in} \tag{46}$$

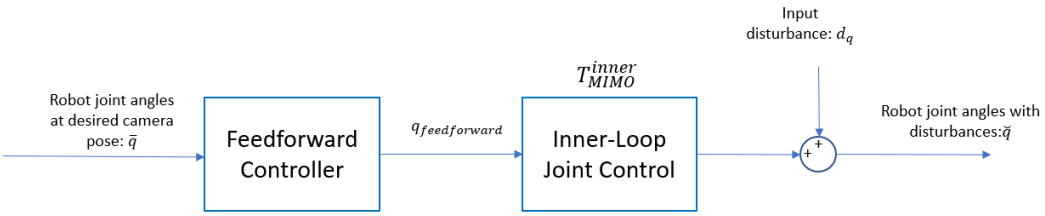


Figure 10. Feedforward control design

6.3. Outer Feedback Control Loop

In control system block diagram (Figure 7), the linear closed loop transfer function has been developed in equation (44) and the eye-in-hand kinematic model is a nonlinear map defined in Equations (21) and (22).

When disturbed joint angles $\check{q}$ are inputs to the eye-in-hand model, the outputs are expressed as $\widehat{p_{image}}$. Revise Equation (22) accordingly, we have:

$$\widehat{p_{image}} = \mathcal{F}(\check{q}(t), Pa, P^O) \tag{47}$$

Where $P^O \epsilon \mathbb{R}^{9X1}$, are fiducial markers' coordinates in inertial frame, and Pa are constant parameters, which consist of camera intrinsic parameters, robot geometric parameters, and transformation matrix between the end-effector to the camera.

By choosing a set of linearized points $\check{q}^0 \epsilon \mathbb{R}^{6X1}$, the model expressed in Equation (47) can be linearized with those points in Jacobian matrix form as:

$$\widehat{p_{image}} = J(\check{q}^0, Pa, P^O)\check{q}(t) + \mathcal{F}(\check{q}^0, Pa, P^O) \tag{48}$$

Where $J(\check{q}^0, Pa, P^O) \epsilon \mathbb{R}^{6X6}$ is the Jacobian matrix of $\mathcal{F}(\check{q}(t), Pa, P^O)$ evaluated as $\check{q} = \check{q}^0$. Assuming $C_1 = J(\check{q}^0, Pa, P^O)$, $C_2 = \mathcal{F}(\check{q}(t), Pa, P^O)$, therefore, Equation (48) can be rewritten as:

$$\widehat{p_{image}} = C_1\check{q}(t) + C_2 \tag{49}$$

Let's define $\widehat{p_{image}}' = \widehat{p_{image}} - C_2$, then, the overall block diagram of the linearized system is shown in Figure 11.

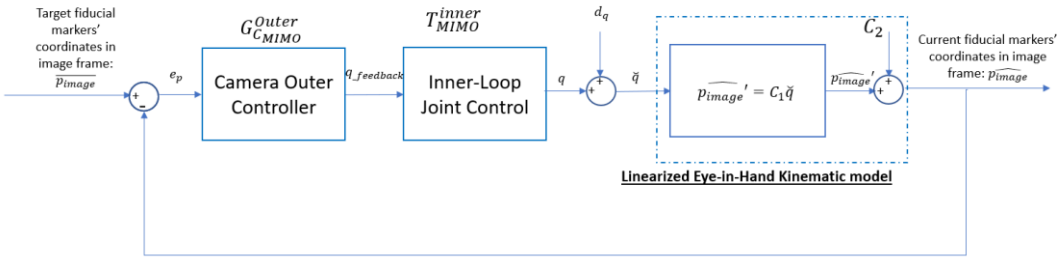


Figure 11. Feedback loop with linearized model.

The linearized plant transfer function is derived as:

$$G_{p_{MIMO\_outer}}^{linear} = \frac{\widehat{p_{image}}'}{q_{feedback}} = C_1 \frac{(3\tau_{in}s + 1)}{(\tau_{in}s + 1)^3} \cdot I_{6\times6} \tag{50}$$

 doi:10.20944/preprints202502.1603.v1

As C_1 is coupled, the first step to derive an observer for the multivariable system using model linearization is to find the Smith-McMillan form of the plant [32].

To get Smith-McMillan form, we can decompose $G_{p_{MIMO\_outer}}^{linear} \epsilon \mathbb{R}^{9X6}$ with singular value decomposition (SVD) as:

$$G_{p_{MIMO\_outer}}^{linear} = U_L M_p U_r \tag{51}$$

where $U_L \epsilon \mathbb{R}^{9X9}$ and $U_r \epsilon \mathbb{R}^{6X6}$ are the left and right unimodular matrices, and $M_p \epsilon \mathbb{R}^{9X6}$ is the Smith-McMillan form of $G_{p_{MIMO\_outer}}^{linear}$.

M_p is a diagonalized transfer function with each nonzero entry equals to a gain multiple the transfer function $\frac{(3\tau_{in}s+1)}{(\tau_{in}s+1)^3}$; For the i^{th} row of M_p the entry on the diagonal is:

$$M_p(i,i) = gain(i) \cdot \frac{(3\tau_{in}s+1)}{(\tau_{in}s+1)^3}, i\epsilon(1,2,3,4,5,6) \tag{52}$$

Where $gain \epsilon \mathbb{R}^{6X1}$ is a numerical vector.

The design of a Youla controller for each nonzero entry in M_p is trivial in this case as all poles/zeros of the plant transfer function are in the left half-plane, and therefore, they are stable. In this case, the selected decoupled Youla transfer function: M_Y can shape the decoupled closed loop transfer function, M_T, by manipulating poles and zeros. All poles and zeros in the original plant can be cancelled out and new poles and zeros can be added to shape the closed-loop system. Let's select a Youla transfer function so that the decoupled closed-loop SISO system behaves like a second order Butterworth filter, such that:

$$M_T = \frac{{\omega_n}^2}{(s^2+2\zeta\omega_n s+{\omega_n}^2)} \cdot \begin{bmatrix} I_{6\times6} & Zero_{3\times3} \\ Zero_{3\times3} & Zero_{3\times3} \end{bmatrix}, \tag{53}$$

where ω_n is called natural frequency and approximately sets the bandwidth of the closed –loop system. It must be ensured that the bandwidth of the outer-loop is smaller than the inner-loop, i.e., $1/\omega_n > \tau_{in}$. ζ is called the damping ratio, which is another tuning parameter. $Zero_{3\times3}$ is a 3×3 matrix with all entries equaling to zero. Note that the coordinates from the last point cannot be controlled in the feedback loop.

Then we can compute the decoupled diagonalized Youla transfer functions $M_Y \epsilon \mathbb{R}^{6X9}$. The diagonal entry of i^{th} row is denoted as $M_Y(i,i)$:

$$M_Y(i,i) = \frac{M_T(i,i)}{M_p(i,i)} = \frac{1}{gain(i)} \frac{{\omega_n}^2}{(s^2+2\zeta\omega_n s+{\omega_n}^2)} \frac{(\tau_{in}s+1)^3}{(3\tau_{in}s+1)}, , i\epsilon(1,2,3,4,5,6) \tag{54}$$

Similar to Equations (40)–(42), the final coupled Youla, closed loop, sensitivity, and observer transfer function matrices are computed as:

$$Y_{MIMO\_outer}^{linear} \epsilon \mathbb{R}^{6X9} = U_R M_Y U_L \tag{55}$$

$$T_{y_{MIMO\_outer}}^{linear} \epsilon \mathbb{R}^{9X9} = G_{p_{MIMO\_outer}}^{linear} \cdot Y_{MIMO\_outer}^{linear} \tag{56}$$

$$S_{y_{MIMO\_outer}}^{linear} \epsilon \mathbb{R}^{9X9} = 1 - T_{y_{MIMO\_outer}}^{linear} \tag{57}$$

$$G_{C_{MIMO\_outer}}^{linear} \epsilon \mathbb{R}^{6X9} = Y_{MIMO\_outer}^{linear} \cdot (S_{y_{MIMO\_outer}}^{linear})^{-1} \tag{58}$$

The controller developed in the above section is based on the linearization of the combined model at a particular linearized point $\breve{q}^0$. This controller can only stabilize at certain range of joint angles around $\breve{q}^0$. As current joint angles $\breve{q}^0$ deviates from $\breve{q}$, the error between the estimated linearized system (48) and the true nonlinear system (47) increases.

To tackle this problem, we develop an adaptive controller that is computed online based on linearization of the model at current joint angles. This control process is depicted in Figure 12.

The first step is to estimate current joint angles $\tilde{q}$ from current measured images coordinates $\widehat{p_{image}}$. The mathematic models of eye-in-hand kinematic model been given in expression is defined

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 February 2025 doi:10.20944/preprints202502.1603.v1

in Equation (22). Therefore, the mathematical function of the inverse model can be derived and expressed as:

$$\tilde{q} = \mathcal{F}^{-1}(\widehat{p_{image}}, Pa, P^O) \tag{59}$$

Where $\mathcal{F}^{-1}$ is the inverse process of Equation (22), which is a combination of coordinate system transformation from the image frame to the end-effector frame and robot inverse kinematics process. Given estimated current angle $\tilde{q}$, we can calculate the Jacobian matrix of the nonlinear model at current time. By obtaining left and right unimodular matrices and Smith-McMillan form from singular value decomposition, the current linear controller $G_{C_{MIMO\_outer}}^{linear}$ can be built by Equations (55)-(58).

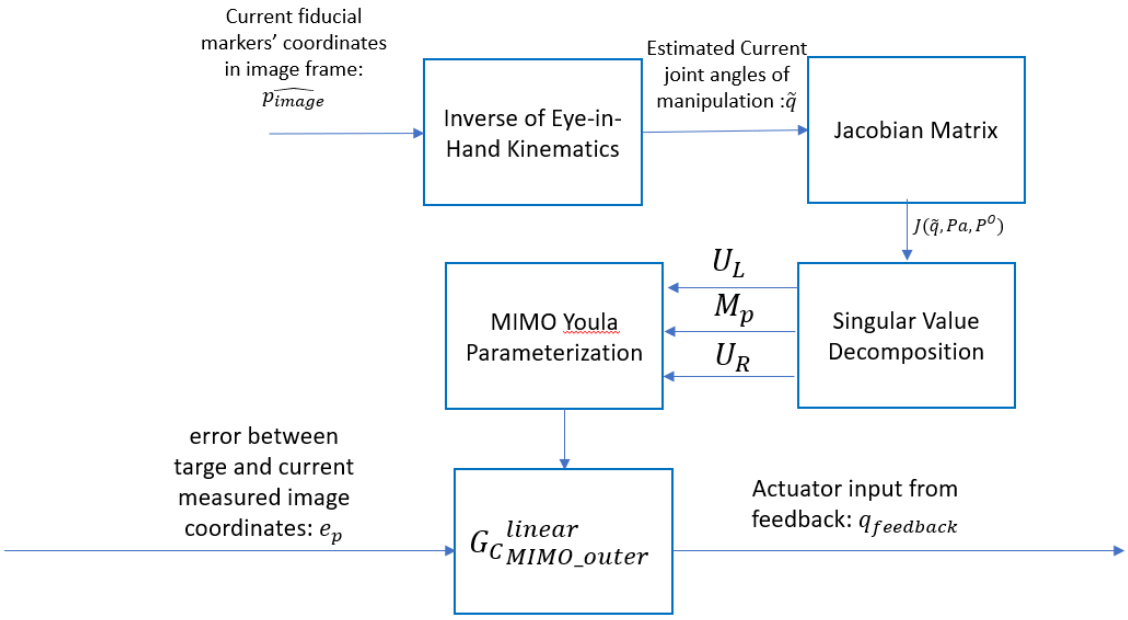


Figure 12. Adaptive feedback loop.

7. Simulations Results

To evaluate the performance of our controller design, we simulated two scenarios in MATLAB Simulink using a Zed 2 stereo camera system [33] and an ABB IRB 4600 elbow robotic manipulator [30]. The specifications for the camera system and robot manipulator are summarized in the tables in the appendix. The camera system performs 2D feature estimation of three virtual points in space, with their coordinates in the inertial frame selected as: $[\text{-}0.5m\ 0\ 0]^T$, $[0\ 0\ 0.5m]^T$, $[2m, \text{-}2\text{m}, 0]^T$.

Many camera noise removal algorithms have been proposed and shown to be effective in practical applications, such as spatial filters [34], wavelet filters [35], and the image averaging technique [36]. Among these denoising methods, there is always a tradeoff between computational efficiency and performance. For this paper, we assume that the images captured by the camera have been preprocessed using one of these methods, and the noise has been almost perfectly attenuated. In other words, the only remaining disturbances in the system are due to unmodeled joint dynamics, such as compliance and flexibility, which are modeled as input disturbances in the controlled system.

Two scenarios were simulated:

- Scenario 1: Without input disturbances.
- Scenario 2: With a 1° step input disturbance added to each joint of the robot arms for the entire simulation time.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 February 2025 doi:10.20944/preprints202502.1603.v1

In both scenarios, the camera system starts from an initial pose in the inertial frame, denoted as $Pose^{O}_{initila}$, and maneuvers target pose, denoted as $Pose^{O}_{final}$. Table 1 summarizes the initial and final poses for each scenario, along with the corresponding joint configurations.

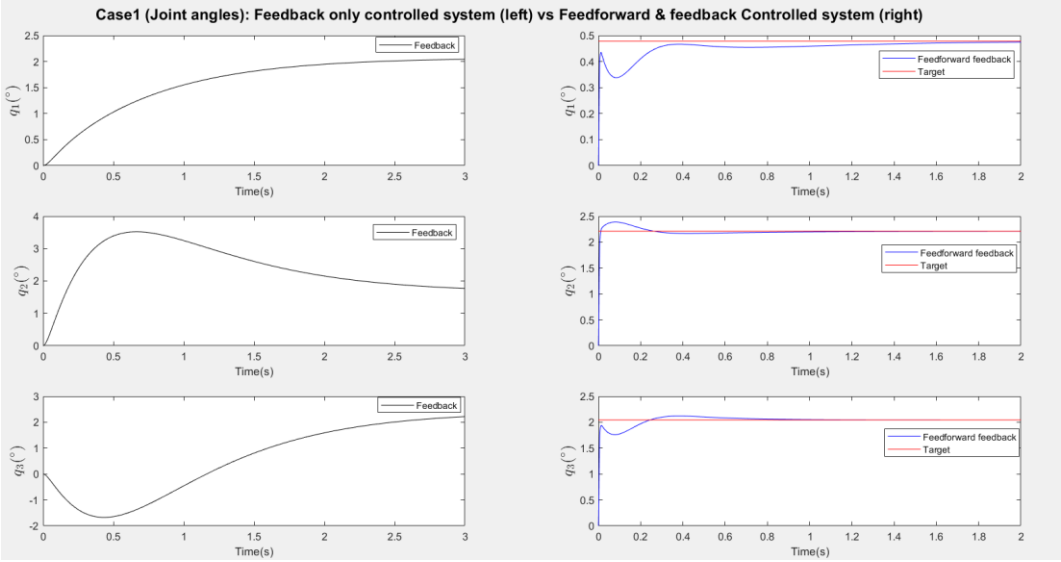
Figures 13 and 15 present the responses of the six joint angles for the two scenarios, respectively. Figures 14 and 16 show the responses of the nine image coordinates over time for each scenario. In each case, the feedback-only controlled system (left plot) is compared to the feedforward-and-feedback controlled system (right plot). These comparisons focus on overshoot, response time, and target tracking performance.

Both scenarios are simulated with bandwidth of the inner-loop as $100 rad/s$ and the bandwidth of the outer-loop as $10 rad/s$.

Table 1. Camera Pose and Robot Joint Angles at Initial and Final State of Simulation Scenarios.

Format	Camera Pose	Robot Joint Angles
	$\begin{bmatrix} n_x & s_x & a_x & d_x \\ n_y & s_y & a_y & d_y \\ n_z & s_z & a_z & d_z \end{bmatrix}$ Where $[n_x, n_y, n_z]^T$, $[s_x, s_y, s_z]^T$ and $[a_x, a_y, a_z]^T$ are Yaw, Pitch, and Roll, and $[d_x, d_y, d_z]^T$ (in meters) is the position, measured in inertial frame {O}.	$\begin{bmatrix} q_1 & q_2 \\ q_3 & q_4 \\ q_5 & q_6 \end{bmatrix}$ Where $[q_1, q_2, q_3, q_4, q_5, q_6]$ in (degrees) are robot joint angles.

	***Scenario* 1**		***Scenario* 2**	
	Camera Pose	Robot Joint Angles	Camera Pose	Robot Joint Angles
Initial State	$\begin{bmatrix} 0 & 0 & 1 & 1.27 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1.57 \end{bmatrix}$	$\begin{bmatrix} 0° & 0° \\ 0° & 0° \\ 0° & 0° \end{bmatrix}$	$\begin{bmatrix} -0.070 & -0.998 & 0.002 & 1 \\ -0.996 & 0.070 & -0.060 & 0 \\ -0.060 & -0.007 & -0.998 & 0 \\ 0 & 0 & 0 & \end{bmatrix}$	$\begin{bmatrix} 42.40° & 21.20° \\ 4.58° & -2.86° \\ 66.46° & -42.40 \end{bmatrix}$
Final State	$\begin{bmatrix} -0.11 & 0.14 & 0.98 & 1. \\ -0.09 & 0.98 & -0.15 & - \\ -0.99 & -0.10 & -0.09 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.48° & 2.21° \\ 2.05° & -82.68 \\ 9.46° & 77.47° \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 & 1 \\ -1 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 45° & 18.59° \\ 4.35° & 0° \\ 67.06° & -45° \end{bmatrix}$



 doi:10.20944/preprints202502.1603.v1

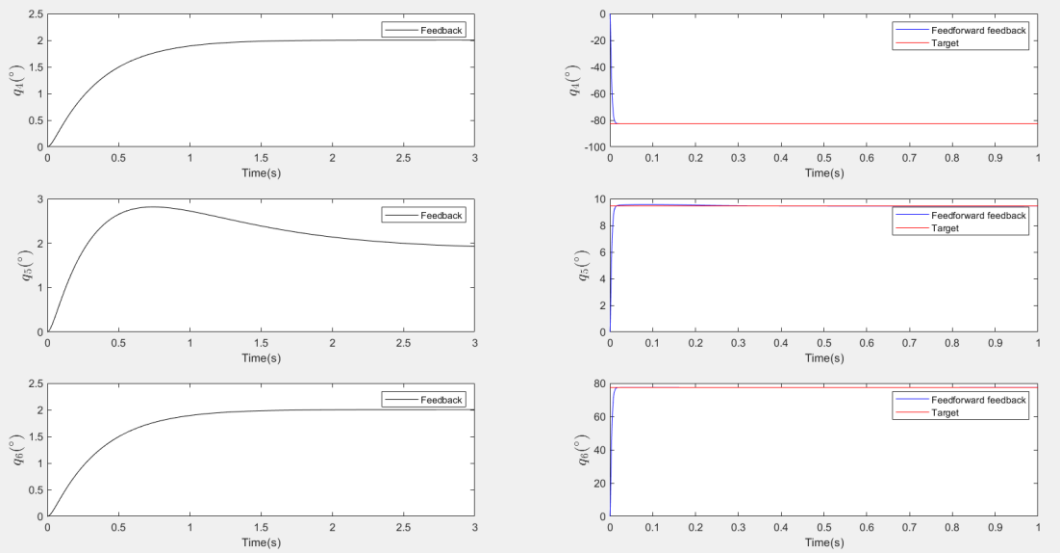
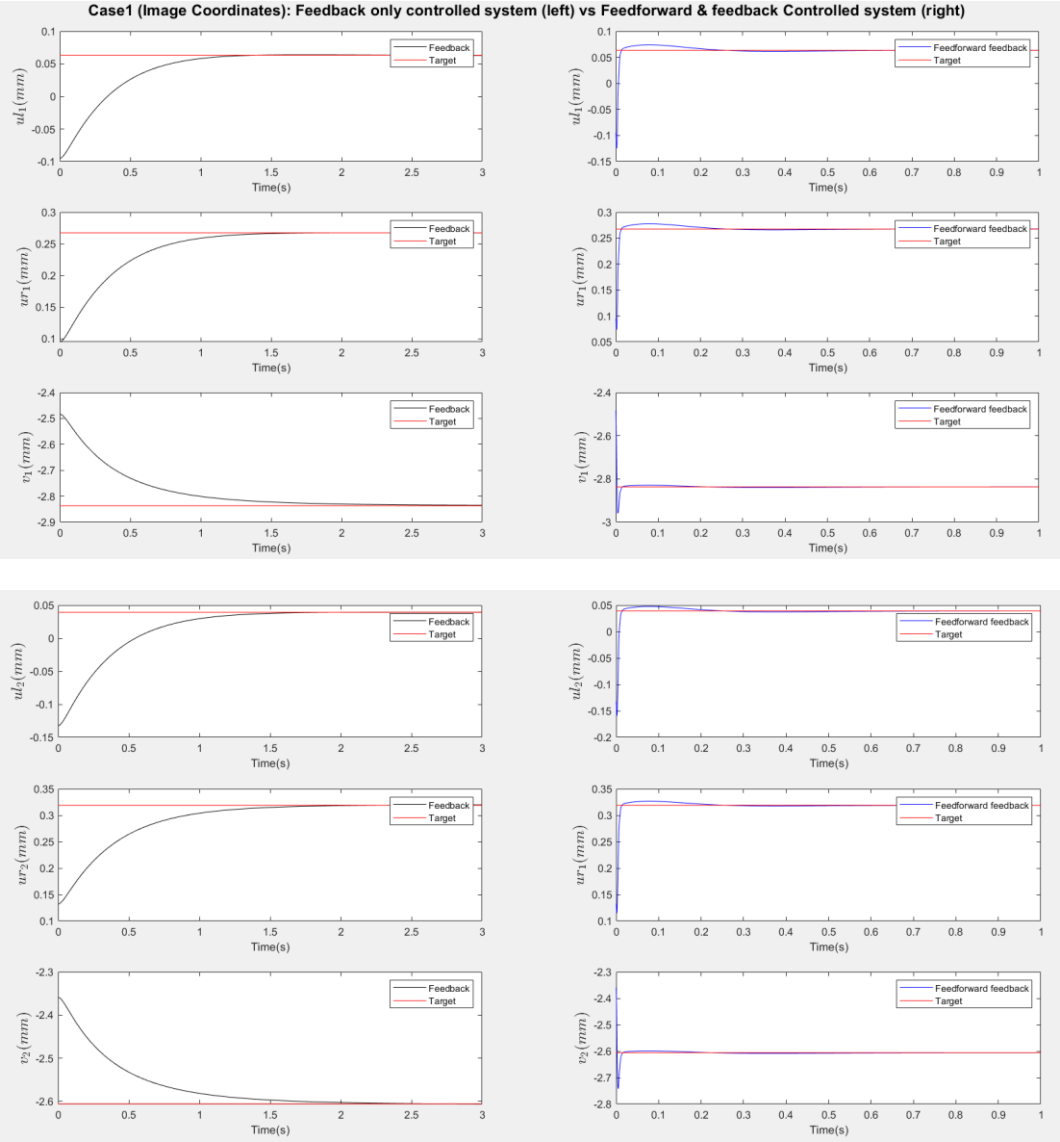


Figure 13. Scenario one (no disturbances): Response of robot joint angles. (*Left: Feed-back only responses, Right: Feedforward-feedback responses).*



 doi:10.20944/preprints202502.1603.v1

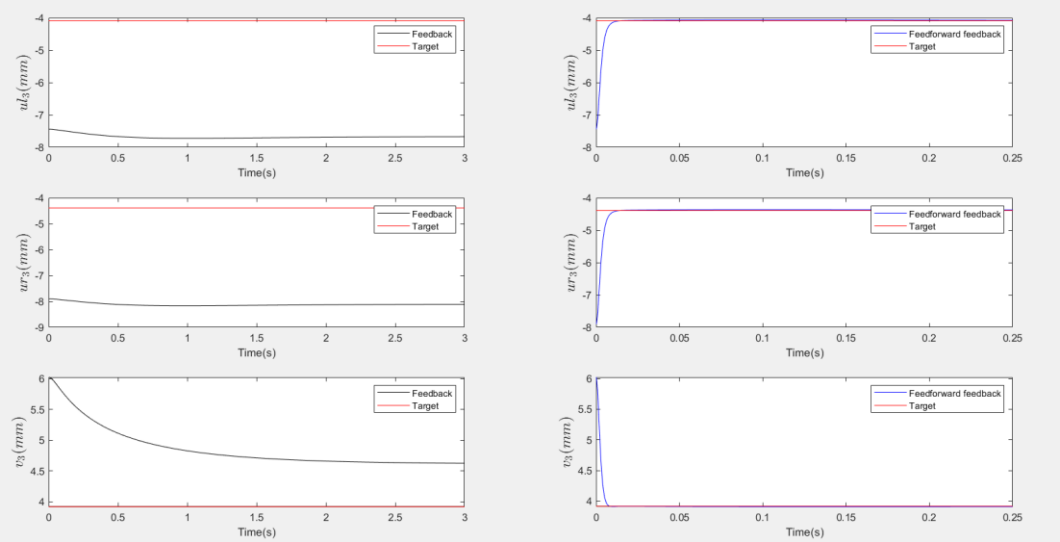
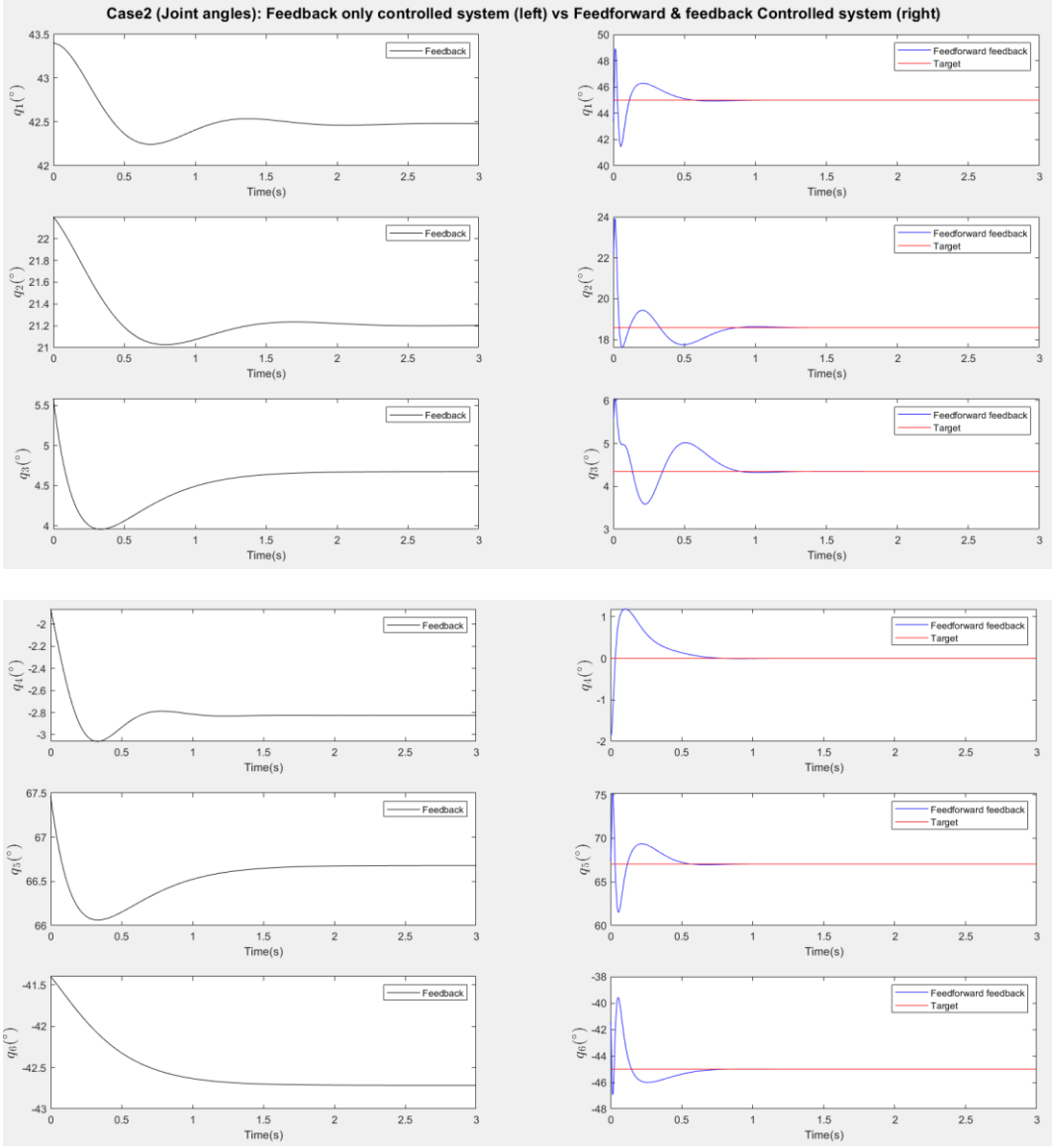
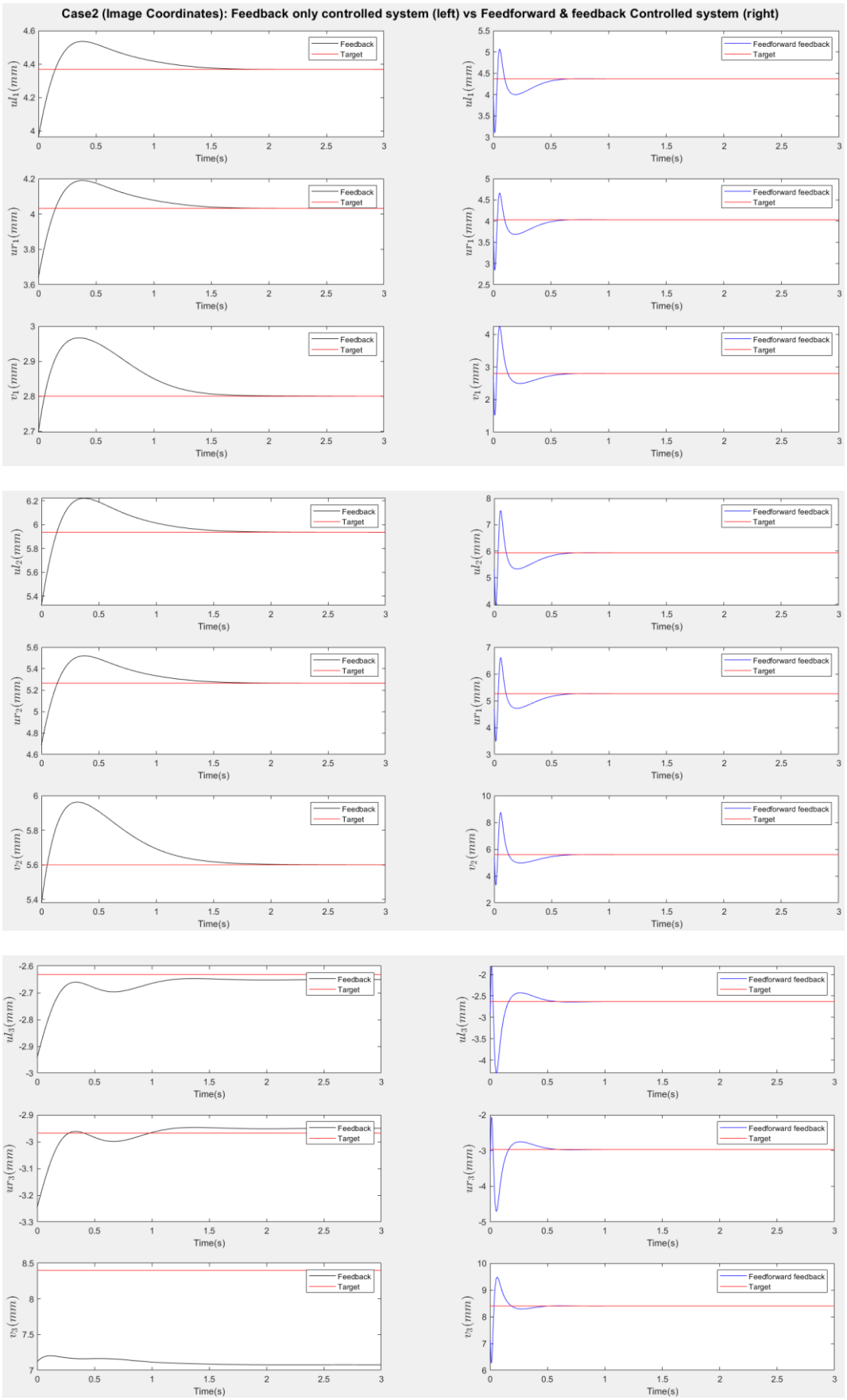


Figure 14. Scenario one (no disturbances): Response of image coordinates. (*Left: Feed-back only responses, Right: Feedforward-feedback responses).* The three coordinates of the third point are only matched in the feedforward-feedback approach.



 doi:10.20944/preprints202502.1603.v1

Figure 15. Scenario two (add disturbances): Response of robot joint angles. (*Left: Feed-back only responses, Right: Feedforward-feedback responses).*



 doi:10.20944/preprints202502.1603.v1

Figure 16. Scenario two (add disturbances): Response of image coordinates. (*Left: Feed-back only responses, Right: Feedforward-feedback responses).* The three coordinates of the third point are only matched in the feedforward-feedback approach.

The response plots indicate that both the feedback-only controller and the combined feedforward-and-feedback controller successfully stabilize the system and reach a steady state within three seconds, even in the presence of small input disturbances (Scenario Two). However, the feedback-only controller fails to guide the camera to its desired pose and falls into local minima, as evident from the third point's coordinates ($ul_{3,}\ ur_{3,}\ v_{3,}$), which do not match the target at steady state. This issue arises from the overdetermined nature of the stereo-based visual servoing system, where the number of output constraints (9) exceeds the degrees of freedom (DoFs) available for control (6). As a result, the feedback controller can only match six out of nine image coordinates, leaving the rest coordinates unmatched.

In contrast, the feedforward-and-feedback controller avoids local minima and accurately moves the camera to the target pose. This is because the feedforward component directly controls the robot's joint angles rather than image features. Since the joint angles (6 DoFs) uniquely correspond to the camera's pose (6 DoFs), the feedforward controller helps the system reach the global minimum by using the desired joint configurations as inputs.

When comparing performance, the system with the feedforward controller exhibits a shorter transient period (less than 2 seconds) compared to the feedback-only system (less than 3 seconds). However, the feedforward controller can introduce overshoot, particularly in the presence of disturbances. This occurs because feedforward control provides an immediate control action based on desired setpoints, resulting in significant initial actuator input that causes overshoot. Additionally, a feedforward-only system is less robust against disturbances and model uncertainties. Fine-tuning the camera's movement under these conditions requires a feedback controller.

Therefore, the combination of feedforward and feedback control ensures fast and accurate camera positioning. The feedforward controller enables rapid convergence toward the desired pose, while the feedback controller improves robustness and corrects errors due to disturbances or uncertainties. Together, they work cooperatively to achieve optimal performance.

8. Conclusions

In this article, we first provide a systematic proof of the PnP problem for a stereo camera system and then propose an innovative control policy to address the overdetermination issues in image-based visual servoing (IBVS) control. Results from two simulation scenarios demonstrate that the proposed algorithm successfully brings the camera to the desired pose with high accuracy and speed.

Several existing approaches [21–23], mentioned in the introduction, have also addressed the issue of local minima in IBVS. Compared to those methods, the key advantage of our system is its simplicity and ease of implementation. A linear feedforward controller is sufficient to handle the local minimum problem without requiring complex online optimization as in MPC or additional 3D feature measurements as in 2 ½-D visual servoing. The feedback loop is designed following the traditional IBVS structure but incorporates a higher-fidelity dynamic model. The adaptive features in the feedback controller stabilize the system across the entire state space, achieved by combining multiple linear Youla-parameterized controllers. To reduce computational overhead, we can lower the online update frequency or predesign several linear Youla controllers offline and switch between them smoothly using a switching algorithm.

However, the feedforward design introduces challenges, particularly with large overshoots that can cause erratic joint movements. This may increase the risk of accidents and potential damage to the robot. A possible solution is to optimize the controller parameters—such as bandwidth and damping ratios—which can be explored in future work. In addition, feedback and feedforward controllers can be designed simultaneously using H_∞ control techniques [37], which optimizes the system's stability and performance in the presence of disturbances.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 February 2025 doi:10.20944/preprints202502.1603.v1

In summary, this paper investigates the overdetermination problem in stereo-based IBVS tasks. While future improvements to the algorithm are possible, the proposed control policy has demonstrated significant potential as an accurate and fast solution for real-world eye-in-hand (EIH) visual servoing tasks.

Appendix A

In this section, we will show the geometric model of a specific robot manipulator ABB IRB 4600 45/2.05[23] and a figure of a camera model: Zed 2 with dimensions [24]. This section also contains specification tables of robots' dimensions, camera, and motor installed inside the joints of manipulators.

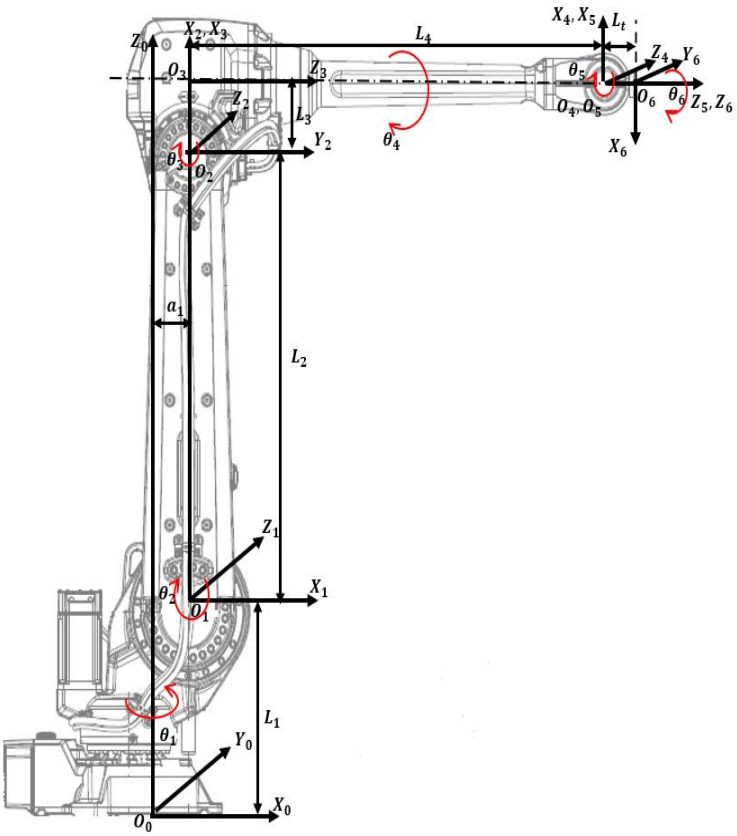
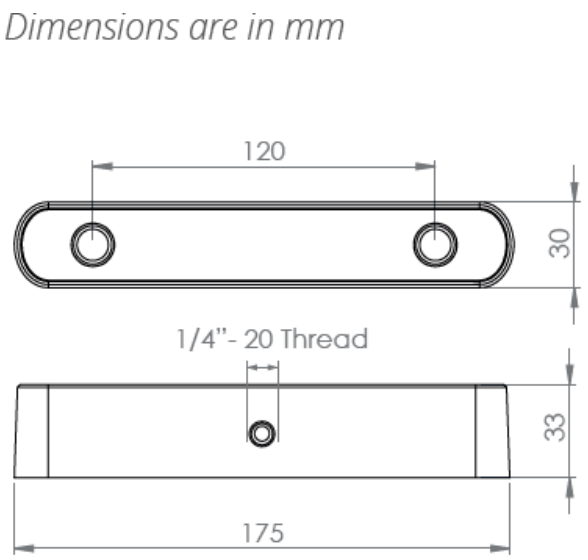


Figure A1. IRB ABB 4600 Model with attached frames.



 doi:10.20944/preprints202502.1603.v1

Figure A2. Zed 2 stereo camera model with dimensions.

Table A1. Specification Table of ABB IRB 4600 45/2.05 Model (Dimensions).

Parameters	**Values**
Length of Link 1: L_1	495 mm
Length of Link 2: L_2	900 mm
Length of Link 3: L_3	175 mm
Length of Link 3: L_4	960 mm
Length of Link 1 offset: a_1	175 mm
Length of Spherical wrist: L_t	135 mm
Tool length (screwdriver): $\overline{PJ}_{tool}$	127 mm

Table A2. Specification Table of ABB IRB 4600 45/2.05 Model (Axis Working range).

Axis Movement	**Working range**
Axis 1 rotation	+180° to -180°
Axis 2 arm	+150° to -90°
Axis 3 arm	+75° to -180°
Axis 4 wrist	+400° to -400°
Axis 5 bend	+120° to -125°
Axis 6 turn	+400° to -400°

Table A3. Specification Table of Stereo Camera Zed 2.

Parameters	**Values**
Focus length: f	2.8 mm
Baseline: B	120 mm
Weight: W	170g
Depth range:	0.5m-25m
Diagonal Sensor Size:	6mm
Sensor Format:	16:9
Sensor Size: W X H	5.23mm X 2.94mm
Angle of view in width: α	*86.09°*
Angle of view in height: β	55.35°

Table A4. Specification Table of Motors and gears.

Parameters	**Values**
DC Motor	
Armature Resistance: R	0.03 Ω
Armature Inductance: L	0.1 mH
Back emf Constant: K_b	7 mv/rpm
Torque Constant: K_m	0.0674 N/A
Armature Moment of Inertia: J_a	0.09847 kgm^2
Gear	
Gear ratio: r	200:1
Moment of Inertia: J_g	0.05 kgm^2
Damping ratio: B_m	0.06

Appendix B

In this section, we will show forward kinematics and inverse kinematics of the 6 DoFs revolute robot manipulators. The results are consistent with the model ABB IRB 4600.

 doi:10.20944/preprints202502.1603.v1

Forward kinematics refers to the use of kinematic equations of a robot to compute the position of the end-effector from specified values for the joint angles and parameters. The equations are summarized in the below:

$$\begin{aligned} n_x &= c_1 s_{23}(s_4 s_6 - c_4 c_5 c_6) - s_1(s_4 c_5 c_6 + c_4 s_6) - c_1 c_{23} s_5 c_6 \\ n_y &= s_1 s_{23}(s_4 s_6 - c_4 c_5 c_6) + c_1(s_4 c_5 c_6 + c_4 s_6) - s_1 c_{23} s_5 c_6 \\ n_z &= c_{23}(s_4 s_6 - c_4 c_5 c_6) + s_{23} s_5 c_6 \end{aligned} \tag{B1}$$

$$\begin{aligned} s_x &= c_1 s_{23}(s_4 c_6 + c_4 c_5 c_6) + s_1(s_4 c_5 s_6 - c_4 c_6) + c_1 c_{23} s_5 s_6 \\ s_y &= s_1 s_{23}(s_4 c_6 + c_4 c_5 c_6) - c_1(s_4 c_5 s_6 - c_4 c_6) + s_1 c_{23} s_5 s_6 \\ s_z &= c_{23}(s_4 c_6 + c_4 c_5 c_6) - s_{23} s_5 s_6 \end{aligned} \tag{B2}$$

$$\begin{aligned} a_x &= -c_1 s_{23} c_4 s_5 - s_1 s_4 s_5 + c_1 c_{23} c_5 \\ a_y &= -s_1 s_{23} c_4 s_5 + c_1 s_4 s_5 + s_1 c_{23} c_5 \\ a_z &= c_{23} c_4 s_5 - s_{23} c_5 \end{aligned} \tag{B3}$$

$$\begin{aligned} d_x &= L_t(-c_1 s_{23} c_4 s_5 - s_1 s_4 s_5 + c_1 c_{23} c_5) + c_1(L_3 s_{23} + L_2 s_2 + a_1) \\ d_y &= L_t(-s_1 s_{23} c_4 s_5 + c_1 s_4 s_5 + s_1 c_{23} c_5) + s_1(L_3 s_{23} + L_2 s_2 + a_1) \\ d_z &= L_t(c_{23} c_4 s_5 - s_{23} c_5) + L_3 c_{23} + L_2 c_2 + L_1 \end{aligned} \tag{B4}$$

$$\begin{aligned} &\text{Note: } c_i \equiv \cos(q_i),\ s_i \equiv \sin(q_i) \\ &c_{i,j} \equiv \cos(q_i + q_j),\ s_{i,j} \equiv \sin(q_i + q_j) \\ &i, j \in \{1,2,3,4,5,6\} \end{aligned} \tag{B5}$$

where $[n_x, n_y, n_z]^T$, $[s_x, s_y, s_z]^T$ and $[a_x, a_y, a_z]^T$ are the end-effector's directional vector of Yaw, Pitch and Roll in base frame $O_0X_0Y_0Z_0$ (Figure A1). And $[d_x, d_y, d_z]^T$ are the vector of absolute position of the center of the end-effector in base frame $O_0X_0Y_0Z_0$. For a specific model ABB IRB 4600-45/2.05 (Handling capacity: 45 kg/ Reach 2.05m) the dimensions and mass are summarized in Table A1.

Inverse kinematics refers to the mathematical process of calculating the variable joint angles needed to place the end-effector in a given position and orientation relative to the inertial base frame. The equations are summarized in the below:

$$\begin{aligned} p_x &= d_x - L_t a_x \\ p_y &= d_y - L_t a_y \\ p_z &= d_z - L_t a_z \end{aligned} \tag{B6}$$

$$q_1 = arctan(\frac{p_y}{p_x}) \tag{B7}$$

$$\begin{aligned} q_2 = \frac{pi}{2} - \arccos\left(\frac{{L_2}^2 + \left(\sqrt{{p_x}^2 + {p_y}^2} - a_1\right)^2 + (p_z - L_1)^2 - {L_3}^2 - {L_4}^2}{2L_2\sqrt{{L_3}^2 + {L_4}^2}}\right) \\ - arctan\left(\frac{p_z - L_1}{\sqrt{{p_x}^2 + {p_y}^2} - a_1}\right) \end{aligned} \tag{B8}$$

$$\begin{aligned} q_3 = \pi - \arccos\left(\frac{{L_2}^2 + {L_3}^2 + {L_4}^2 - \left(\sqrt{{p_x}^2 + {p_y}^2} - a_1\right)^2 - (p_z - L_1)^2}{2L_2\sqrt{{L_3}^2 + {L_4}^2}}\right) \\ - \arctan\left(\frac{L_4}{L_3}\right) \end{aligned} \tag{B9}$$

$$q_5 = \arccos(c_1 c_{23} a_x + s_1 c_{23} a_y - s_{23} a_z) \tag{B10}$$

 doi:10.20944/preprints202502.1603.v1

$$q_4 = \arctan\left(\frac{s_1 a_x - c_1 a_y}{c_1 s_{23} a_x + s_1 s_{23} a_y + c_{23} a_z}\right) \tag{B11}$$

$$q_6 = -\arctan\left(\frac{c_1 c_{23} s_x + s_1 c_{23} s_y - s_{23} s_z}{c_1 c_{23} n_x + s_1 c_{23} n_y - s_{23} n_z}\right) \tag{B12}$$

$$\begin{gathered}\text{Note: } c_i \equiv \cos(q_i),\ s_i \equiv \sin(q_i) \\ c_{i,j} \equiv \cos(q_i + q_j),\ s_{i,j} \equiv \sin(q_i + q_j) \\ i, j \in \{1,2,3,4,5,6\}\end{gathered} \tag{B13}$$

where $[n_x, n_y, n_z]^T$, $[s_x, s_y, s_z]^T$, $[a_x, a_y, a_z]^T$ and $[d_x, d_y, d_z]^T$ have been defined above in the forward kinematic discussion.

References

1. E. Cai, R. Rossi, and C. Xiao, "Improving learning-based camera pose estimation for image-based augmented reality applications," in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, Hamburg, Germany, 2023, pp. 1-6. doi: 10.1145/3544549.3585756.
2. Stier, N., Angles, B., Yang, L., Yan, Y., Colburn, A., & Chuang, M. (2023). *LivePose: Online 3D Reconstruction from Monocular Video with Dynamic Camera Poses*. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.
3. X. Li and H. Ling, "Hybrid Camera Pose Estimation With Online Partitioning for SLAM," in *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1453–1460, Apr. 2020. doi: 10.1109/LRA.2020.2967688.
4. S. S. Jacob and S. S, "A comparative study of pose estimation algorithms for visual navigation in autonomous robots," *International Robotics & Automation Journal*, vol. 9, no. 3, pp. 1–7, 2023. doi: 10.15406/iratj.2023.09.00343.
5. T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, and S. Birchfield, "Camera-to-Robot Pose Estimation from a Single Image," arXiv preprint arXiv:1911.09231, 2020. [Online]. Available: https://arxiv.org/abs/1911.09231
6. *Fischler, M. A.; Bolles, R. C. (1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Communications of the ACM.* **24** *(6): 381–395. doi:10.1145/358669.358692. S2CID 972888.*
7. M. Bujnak, Z. Kukelova and T. Pajdla, "A general solution to the P4P problem for camera with unknown focal length," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008, pp. 1-8, doi: 10.1109/CVPR.2008.4587793.
8. Lepetit, V.; Moreno-Noguer, M.; Fua, P. (2009). "EPnP: An Accurate O(n) Solution to the PnP Problem". *International Journal of Computer Vision*. **81** (2): 155–166. doi:10.1007/s11263-008-0152-6. hdl:2117/10327. S2CID 207252029
9. Terzakis, George; Lourakis, Manolis (2020). "A Consistently Fast and Globally Optimal Solution to the Perspective-n-Point Problem". *Computer Vision – ECCV 2020*. Lecture Notes in Computer Science. Vol. 12346. pp. 478–494. doi:10.1007/978-3-030-58452-8_28. ISBN 978-3-030-58451-1. S2CID 226239551
10. M.N. Alkhatib, A.V. Bobkov, N.M. Zadoroznaya, Camera pose estimation based on structure from motion, Procedia Computer Science, Volume 186, 2021, Pages 146-153, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2021.04.205.
11. J. Wang, Y. Wang, C. Guo, S. Xing and X. Ye, "Fusion of Visual Odometry Information for Enhanced Camera Pose Estimation," *2023 8th International Conference on Control, Robotics and Cybernetics (CRC)*, Changsha, China, 2024, pp. 306-309, doi: 10.1109/CRC60659.2023.10488546.
12. Naseer T., Burgard W. Deep regression for monocular camera-based 6-dof global localization in outdoor environments 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE (2017), pp. 1525-1530
13. A. Kendall, M. Grimes, R. Cipolla, Posenet: A convolutional network for real-time 6-DOF camera relocalization, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2938–2946.
14. A. Kendall, R. Cipolla, Geometric loss functions for camera pose regression with deep learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5974–5983.
15. F. Chaumette and S. Hutchinson. Visual servo control. I. Basic approaches. IEEE Robotics & Automation Magazine.2006;13(4):82-90. DOI: 10.1109/MRA.2006.250573.
16. Y. Ma, X. Liu, J. Zhang, D. Xu, D. Zhang, and W. Wu, "Robotic grasping and alignment for small size components assembly based on visual servoing," Int. J. Adv. Manuf. Technol., vol. 106, nos. 11–12, pp. 4827–4843, Feb. 2020.

 doi:10.20944/preprints202502.1603.v1

17. T. Hao, D. Xu and F. Qin, "Image-Based Visual Servoing for Position Alignment With Orthogonal Binocular Vision," in IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1-10, 2023, Art no. 5019010, doi: 10.1109/TIM.2023.3289560.
18. M. Sheckells, G. Garimella and M. Kobilarov, "Optimal Visual Servoing for differentially flat underactuated systems," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea (South), 2016, pp. 5541-5548, doi: 10.1109/IROS.2016.7759815.
19. Wang, Yuanze, et al. "NeRF-IBVS: Visual Servo Based on NeRF for Visual Localization and Navigation." *Advances in Neural Information Processing Systems*, 2023.
20. Guo K, Cao R, Tian Y, Ji B, Dong X, Li X. Pose and Focal Length Estimation Using Two Vanishing Points with Known Camera Position. Sensors (Basel). 2023 Apr 3;23(7):3694. doi: 10.3390/s23073694. PMID: 37050754; PMCID: PMC10098530.
21. N. R. Gans and S. A. Hutchinson, "Stable Visual Servoing Through Hybrid Switched-System Control," in *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 530-540, June 2007, doi: 10.1109/TRO.2007.895067. keywords: {Visual servoing;Cameras;Error correction;Robot vision systems;Gallium nitride;Servosystems;Control
22. F. Chaumette and E. Malis, "2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings," *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, San Francisco, CA, USA, 2000, pp. 630-635 vol.1, doi: 10.1109/ROBOT.2000.844123.
23. P. Roque, E. Bin, P. Miraldo and D. V. Dimarogonas, "Fast Model Predictive Image-Based Visual Servoing for Quadrotors," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2020, pp. 7566-7572, doi: 10.1109/IROS45743.2020.9340759.
24. Zhu, X., Bai, Y., Yu, C. et al. A new computational approach for optimal control of switched systems. *J Inequal Appl* **2024**, 53 (2024). https://doi.org/10.1186/s13660-024-03124-2
25. E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D Visual Servoing," 1999.
26. Z. Ma and J. Su, "Robust uncalibrated visual servoing control based on disturbance observer," *ISA Transactions*, vol. 59, pp. 193–204, 2015. doi: 10.1016/j.isatra.2015.07.003.
27. Youla, D., Jabr, H., and Bongiorno, J. Modern Wiener-Hopf Design of Optimal Controllers-Part II: The Multivariable Case. IEEE Transactions on Automatic Control.1976; 21(3):319-338. DOI: 10.1109/TAC.1976.1101223.
28. Illingworth J., Kittler J.: A survey of the Hough transform. Comput. Vis. Graph. Image Process. 44(1), 87–116 (1988)
29. Denavit, Jacques; Hartenberg, Richard Scheunemann (1955). A kinematic notation for lower-pair mechanisms based on matrices. Trans ASME J. Appl. Mech. .1955;23 (2): 215–221. DOI:10.1115/ 1.4011045.
30. Anonymous. ABB IRB 4600 -40/2.55 Product Manual [Internet]. 2013. Available from: https://www.manualslib.com/manual/1449302
31. Mark, W.S., M.V. (1989). Robot Dynamics and control. John Wiley & Sons, Inc. 1989.
32. F. Assadian and K. Mallon "Robust Control: Youla Parameterization Approach" John Wiley & Sons, Ltd., 2022, ch. 10, pp. 217-246. doi: 10.1002/9781119500292.ch10.
33. Anonymous. Stereolabs Docs: API Reference, Tutorials, and Integration. Available from: https://www.stereolabs.com/docs [Accessed: 2024-7-18]
34. Patidar, P., Gupta, M., Srivastava, S., & Nagawat, A.K. Image De-noising by Various Filters for Different Noise. International Journal of Computer Applications. 2010; 9: 45-50.
35. R. Zhao and H. Cui. Improved threshold denoising method based on wavelet transform. 2015 7th International Conference on Modelling, Identification and Control (ICMIC); 2015: pp. 1-4. DOI: 10.1109/ICMIC.2015.7409352.
36. Ng J., Goldberger J.J. Signal Averaging for Noise Reduction. In: Goldberger J., Ng J. (eds) Practical Signal and Image Processing in Clinical Cardiology. London: Springer; 2010. p. 69-77. DOI: 10.1007/978-1-84882-515-4.ch7
37. Zhou, K., Doyle, J. C., & Glover, K. (1996). *Robust and Optimal Control*. Prentice Hall.