

Article

Not peer-reviewed version

Energy Aware Camera Location Search Algorithm for Increasing Precision of Observation in Automated Manufacturing

[Rongfei Li](#) * and [Farhad Assadian](#) *

Posted Date: 23 August 2024

doi: 10.20944/preprints202408.1718.v1

Keywords: energy-aware; optimization; visual servoing; eye-to-hand configuration; automated manufacturing



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Energy Aware Camera Location Search Algorithm for Increasing Precision of Observation in Automated Manufacturing

Rongfei Li * and Francis Assadian

University of California, Davis; fassadian@ucdavis.edu

* Correspondence: rfli@ucdavis.edu

Abstract: Visual servoing technology has been well developed and applied in many automated manufacturing tasks, especially in tools' pose alignment. To access a full global view of tools, most applications adopt eye-to-hand configuration or eye-to-hand/eye-in-hand cooperation configuration in an automated manufacturing environment. All research papers mainly put efforts into developing control and observation architectures in various scenarios, but none of them has discussed the importance of the camera's location in eye-to-hand configuration. In a manufacturing environment, the quality of camera estimations may vary significantly from one observation location to another, as the combined effects of environmental conditions result in different noise levels of a single image shot at different locations. In this paper, we propose an algorithm for the camera's moving policy so that it explores the camera workspace and searches for the optimal location where the images' noise level is minimized. Also, this algorithm ensures the camera ends up at a suboptimal (if the optimal one is unreachable) location among the locations already searched, with limited energy available for moving the camera. An automated manufacturing application has been simulated and the results show the success of this algorithm's improvement of observation precision with limited energy.

Keywords: energy-aware; optimization; visual servoing; eye-to-hand configuration; automated manufacturing

1. Introduction

In automated industry, automatic alignment plays a critical role in many different applications, such as micromanipulation, autonomous welding, and industrial assembly [1–3]. Visual servoing is the technique commonly used in the tasks of alignment as it can guide robotic manipulators to their desired poses (positions and orientations) [4–6]. Visual servoing is the method of controlling a robot's motion using real-time feedback from visual sensors that continuously extract image features [7].

According to the relative positions between cameras and the tool manipulators, visual servoing can be generally divided into two categories: eye-in-hand (EIH) configuration and eye-to-hand (ETH) configuration [8,9]. In EIH, a camera is mounted directly on a robot manipulator, in which case the motion of the robot induces motion of the camera, while in ETH, the camera is fixed in the workspace and observes the motion of the robot from a stationary configuration. Both configurations have their own merits and drawbacks regarding field of view limit and precision that oppose each other. EIH has a partial but precise view of the scene whereas ETH has a global but less precise view of it [10]. For complex tasks in automated manufacturing, one configuration of visual servoing is not adequate as those tasks require the camera to provide global views and to be maneuverable enough to explore the scene [10]. To take advantage of both stationary and robot-mounted sensors, two configurations can be employed in a cooperative scheme, which can be seen in some of the work [11–13].

However, we haven't seen enough discussion on the importance of cameras' location in the ETH or ETH/EIH cooperative configurations from any work published. To observe a single point in the workspace, there are an infinite number of poses that a fixed camera can be placed in the space. Among all possible poses of the camera that keep the same target within the field of view limit, the

image noise at different locations may play a great role in the precision of observations. Many environmental conditions (such as illumination, temperature, etc.) affect the noise level of a single image [14]. In a manufacturing environment, the quality of estimations from a camera may vary significantly from one observation location to another. For instance, the precision of an estimation improves greatly if the camera moves from a location where it is placed within the shade of machines to a location that has better illumination. Also, places near machines may be surrounded by strong electrical signals that also introduce extra noise into the camera sensors.

It is intuitive to consider applying filters to reduce image noise, and therefore, to increase the precision of observations. The Gaussian white noise can be dealt with using spatial filters, e.g., the Gaussian filter, the Mean filter, and the Wiener filter [15]. In addition, wavelet methods [16,17] have the benefit of keeping more useful details but at the expense of computational complexity. However, filters that operate in the wavelet domain still inevitably filter out (or blur) some important high-frequency useful information of the original image, even though they succeed in preserving more edges, compared with spatial filters. Especially at locations where the signal-to-noise ratio (SNR) in an image is low, no matter what filters are applied, it is difficult to safeguard the edges of the noise-free image as well as reduce the noise to a desirable level. Thus, developing an algorithm that searches for locations of the camera where SNR is high is beneficial.

We can approach the denoising problem with multiple noisy images taken from the same perspective. This method is called signal averaging (or image averaging in the application of image processing) [18]. Assume images taken from the same perspective but at different times have the same noise levels. Then averaging multiple images at the same perspective will reduce the unwanted noise as well as retain all original image details. We can also assume the robot's end effector holds the camera rigidly so that any shaking and drift are negligible when shooting pictures. Moreover, the number of images required for averaging is proportional to the ratio of an image's original noise level to the reduced level that is desirable for observations. Therefore, the number of images required is a measurement of noise level in a single image. Furthermore, in the denoising process, the precise estimations require that the original image details are retained as much as possible. Based on previous statements, we decided to choose image averaging over all other denoising techniques in this work.

In this paper, we propose an algorithm that searches for the camera's workspace to find an optimal location (if its orientation is fixed) of the camera so that a single image taken at this location has the smallest noise level among images taken at all locations in the space. With limited energy for moving the camera, this algorithm also ensures the camera ends up at a suboptimal (if the optimal pose is unreachable) location among the locations already searched.

2. Methods

2.1. Camera Location Search Process and Algorithm Flowchart

The proposed algorithm is aimed to search and move the camera to the location where the least number of pictures are needed to reduce the noise level of the averaged image to a degree $\sigma_{noise_reduced}$. A flowchart describing the algorithm is shown in Figure 1.

A camera, mounted on a 6-joint robot manipulator, moves freely in 6 degrees of freedom (6 DoFs) in space. Assuming the camera is fixed in orientation towards the face of the object of interest, only the position (3 DoFs) of the camera can vary from the movement of the robot arm. The camera can only move to the locations within the maximum reach of the robot manipulator and the object can only be detectable within the view angle of the camera. Those constraints create a camera operational space (Figure 2a), which is then gridded with a certain resolution of distance. All the intersections of grids come up with a set of nodes (S_{total}), which are all possible candidates of locations that the camera may move and search utilizing the algorithm. Each node is indexed and denoted as $Node_{index}$.

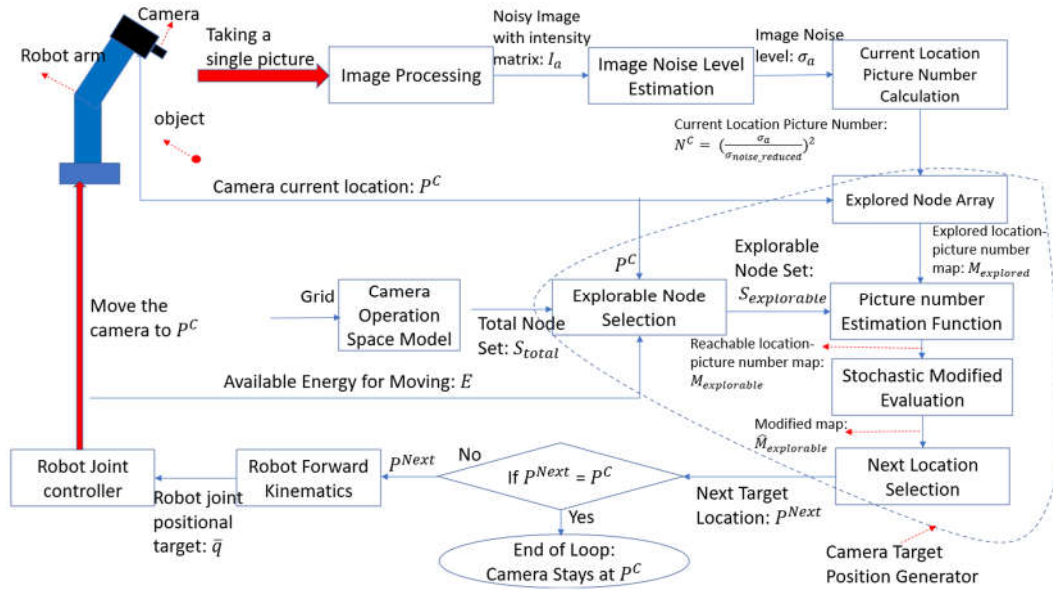


Figure 1. Algorithm flowchart.

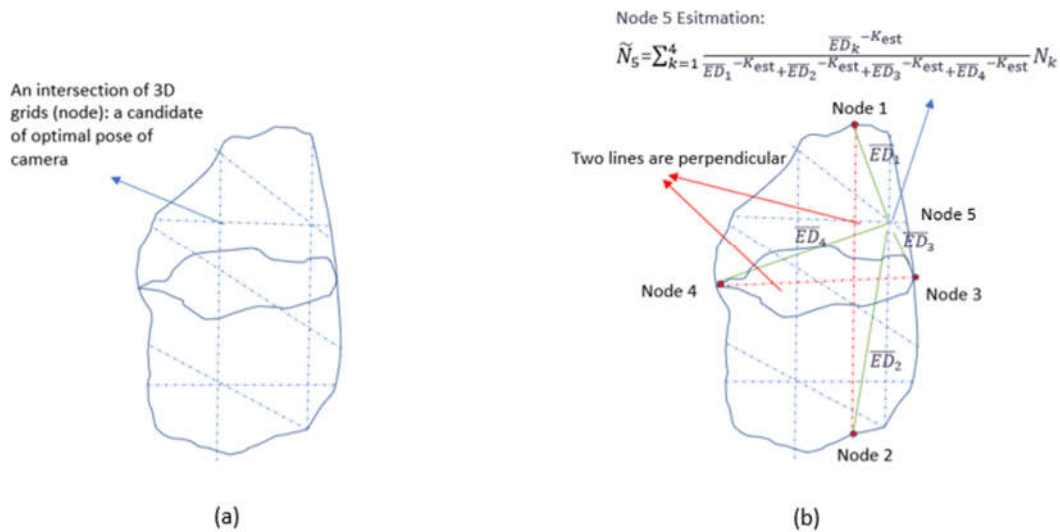


Figure 2. (a): The operational space of the camera is an arbitrary closed 3D geometry in the space shown in the plot. The gridded area is in 3D (blue dashed lines). (b): Four initial nodes are chosen so that their connection lines (red dashed line) are perpendicular. The picture number of Node 5 can be estimated by Equation (1) from four initial explored nodes. \overline{ED} represents Euclidean distance (green line).

The algorithm iteratively commands the camera to take a picture at one location, generate the next target location, and move to that location until it finds the optimal location that requires the least number of pictures. In one iteration, the camera takes a photo at the current location P^C and after image processing, it produces a noisy image with an intensity matrix I_a . A previously developed algorithm [19] estimates the noise level σ_a across the image. Then, at the current location of the camera, the number of pictures N^C needed to reduce the noise level to $\sigma_{noise_reduced}$ is calculated by the equation: $N^C = \left(\frac{\sigma_a}{\sigma_{noise_reduced}}\right)^2$. Based on this information (N^C at P^C), the algorithm then generates the next target location of the camera, P^{Next} . If the next generated target position P^{Next} is the same as the current position P^C , P^C is the optimal location and the camera stays and observes the object at this location. Otherwise, the robot joint controller moves the robot joint angles to the

targets \bar{q} (calculated from forward kinematics [20] (Appendix Equation (B1) -(B5))) so that the camera will move to the next target position, P^{Next} .

2.2. Core Algorithm

The core of this algorithm is to generate the target locations to be explored in the camera's operational space. The details of this algorithm are shown in the area circled with dashed lines in Figure 1. In this section, we will put efforts into the development of the picture number estimation function, the explorable nodes set, the stochastic modified evaluation function, the next location selection process, and the estimated energy cost function.

2.2.1. Picture Number Estimation Function

In each iteration of the algorithm, the camera moves to a new node with location P^C , and the algorithm evaluates the number of pictures N^C needed at this location. The node ($Node_{index}$) is then set to be explored and saved in a set $S_{explored}$. Each node that has been explored in iterations is one-on-one paired with the values: location, and picture numbers of that node. All pairs that are expressed as $Node_{index}^{explored}: (P_{index}^{explored}, N_{index}^{explored})$ are saved in a hash map $M_{explored}$. Then $M_{explored}$ is used to develop an estimation function that can estimate the number of pictures needed for the rest unexplored nodes. The estimated picture number calculated for all nodes in space is denoted as $S_{\tilde{N}}$. Expressions of any element in $S_{\tilde{N}}$ are:

For any node index x in the Total Node Set: $S_{total} = \{Node_x | x \in (1, 2, \dots, m)\}$, where m is the total number of nodes.

$$\tilde{N}_x = \begin{cases} \frac{\sum_{i \in S_{index-exp}} \frac{(\overline{ED}(P_i^{exp}, P_x)^{-K_{est}})}{\sum_{j \in S_{index-exp}} (\overline{ED}(P_j^{exp}, P_x)^{-K_{est}})} N_i^{explored}}{N_x^{explored}}, & \text{if } Node_x \in S_{explored} \\ \text{if } Node_x \notin S_{explored} \end{cases} \quad (1)$$

$$\text{Then: } S_{\tilde{N}} = \{\tilde{N}_x | x \in (1, 2, \dots, m)\} \quad (2)$$

where K_{est} is a positive constant parameter

$$S_{index-exp} = \{index | Node_{index} \in S_{explored}\}, \quad (3)$$

$$\overline{ED}(P_a, P_b) = \sqrt{(P_{ax} - P_{bx})^2 + (P_{ay} - P_{by})^2 + (P_{az} - P_{bz})^2},$$

with P_a, P_b are 3D locations of $Node_a$ and $Node_b$: (4)

$$P_a = (P_{ax}, P_{ay}, P_{az}) \text{ and } P_b = (P_{bx}, P_{by}, P_{bz})$$

Equation (1) is a weighted function concerning each explored node. If one explored node $Node_i^{explored}$ is closer to the node $Node_x^{explored}$, its picture number $N_i^{explored}$ has a larger weight (effect) on the estimation of \tilde{N}_x . Thus, a minus sign of K_{est} is utilized to indicate the negative correlation between the Euclidean distance and weight.

Equation (1) shows that at least some initial nodes are required to be explored before their values can be used to estimate the picture numbers in other nodes. For the estimation function to work properly by covering all nodes in the camera operational space, four initial nodes are selected in the following steps:

1. Find the first two nodes on the boundary of operational space so that the Euclidean distance between these two nodes is the largest.
2. Select other two nodes whose connection line is perpendicular to the previous connection line and the Euclidean distance between these two nodes is also the largest among all possible node pairs.

3. Move the camera to those four nodes in space with proper orientation. Take a single image at each location and estimate the number of pictures required at those locations. And save all four nodes in $S_{explored}$ and their values in $M_{explored}$.

Figure 2b shows the selection of four initial nodes in a camera operational space and shows an example of the use of the estimation function to estimate the picture number in one node.

2.2.2. Explorable Node Selection

A stage in the algorithm, Explorable Node Selection, generates a set of nodes $S_{explorable}$, which contains all the camera's next explorable locations from the current location P^c with available current energy for moving E_{bound} . The explorable set is found in the following steps (Figure 3):

1. With current energy bound E_{bound} , find all nodes in the space that can be reached from the current node P^c .

In other words, all feasible nodes are in the set:

$$S_{feasible} = \{Node_{index} | E(P^c, P_{index}) \leq E_{bound}\} \quad (5)$$

where P is the position of a node and $E(P_a, P_b)$ is the estimated energy cost from P_a to P_b .

When the current energy bound E_{bound} is greater than a predefined energy threshold E_T ($E_{bound} > E_T$), then the algorithm is safe to explore all nodes in the feasible range $S_{feasible}$ in the next iteration loop. In other words:

$$S_{explorable} = S_{feasible}, \quad \text{if } E_{bound} > E_T \quad (6)$$

Note: $E(P_a, P_b)$ is generated from a dynamic model of a robot arm control system. The details of this function are developed in the following section of this paper.

2. When $E_{bound} \leq E_T$, step 2 and step 3 are applied in finding the explorable set. In the feasible set, look at all explored nodes and find the one that has the smallest number of pictures. In other words:

$$S_{feas\&exp} = \{Node_{index} | Node_{index} \in S_{feasible} \cap S_{explored}\} \quad (7)$$

$$exp_min = \{index | N_{exp_min} = \min(N_{index} | Node_{index} \in S_{feas\&exp})\} \quad (8)$$

where $S_{explored}$ is the set of all explored nodes.

$S_{feas\&exp}$ is the intersection set between $S_{feasible}$ and $S_{explored}$.

3. The estimation function may not give accurate results for some unexplored nodes. Therefore, it is possible that the algorithm may make the camera end up in a node that has a large number of pictures in some iteration loops. Because of that, our algorithm needs to make sure at worst it has enough energy to go back to the best (minimum number of pictures) node that has been explored when the available energy is low ($E_{bound} \leq E_T$). This further reduces the feasible set $S_{feasible}$. The explorable set $S_{explorable}$ can be written as:

$$S_{explorable} = \{Node_{index} | E_1 + E_2 \leq E_{bound}\}, \quad \text{if } E_{bound} \leq E_T \quad (9)$$

$$E_1 = E(P^c, P_{index}), E_2 = E(P_{index}, P_{exp_min}) \quad (10)$$

Figure 3 illustrates the selection of explorable nodes set.

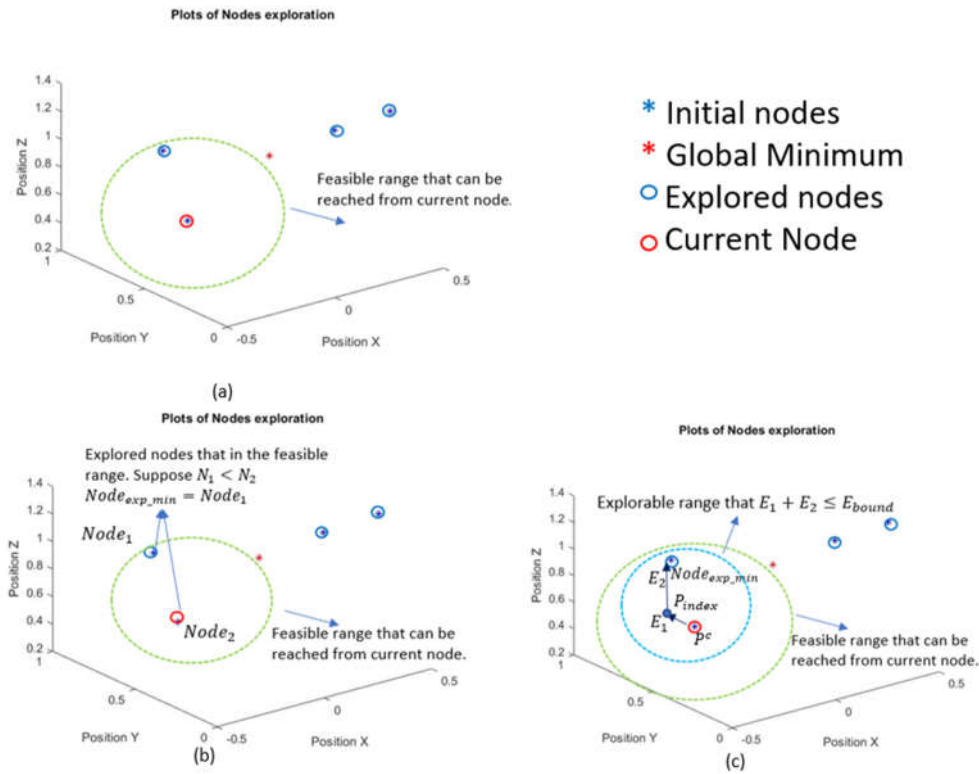


Figure 3. Find the explorable set $S_{explorable}$ when the global minimum (red star) is inaccessible. (a). Step 1: Obtain the feasible range $S_{feasible}$ (green dashed line). (b). Step 2: Find the explored node with a minimum picture number in the feasible range. Initial nodes (blue stars) are explored (blue circles). $Node_1$ is $Node_{exp_min}$. (c). Step 3: Obtain explorable range when $E_{bound} \leq E_T$. The explorable range $S_{explorable}$ (blue dashed line) is a subset of $S_{feasible}$.

2.2.3. Stochastic Modified Evaluation

For all nodes in $S_{explorable}$, we set up a hash map $M_{explorable}$. For each node, its location and number of required pictures for that node are given as: $Node_{index}^{explorable} : (p_{index}^{explorable}, \bar{N}_{index}^{explorable})$. The number of pictures required for nodes in $S_{explorable}$ are calculated from the estimation function in Equation (1).

The algorithm tends to select nodes located around the explored node that has the minimum number of pictures because, from Equation (1), those nodes tend to have the smallest estimated number of pictures. However, this process does not guarantee reaching the global minimum, therefore, minimizing the number of pictures. To sufficiently explore unknown nodes as well as exploit information from already explored nodes, a stochastic process is introduced to modify the picture number estimation function given in Equation (1).

Assume when the locations of a camera in space deviate with a smaller amount from each other, the difference of real picture number values at those locations is also smaller. Thus, among all unexplored nodes, the estimated picture number of an unexplored node, which is closer to the explored node, is more deterministic. To emphasize this feature, we make the estimated picture number follow a normal distribution with a mean $\mu_{\bar{N}}$ being equal to the value from Equation (1) and a standard deviation $\sigma_{\bar{N}}$. As an unexplored node is further away from explored nodes, the larger value of $\sigma_{\bar{N}}$ should be assumed, which means more indeterministic estimation (illustrated in Figure 4). The following equations summarize the above discussion:

For any nodes $Node_{index=x}^{explorable}$ in $S_{explorable}$, its picture number estimation should follow the following normal distributions:

$$P(Z_x) = \mathcal{N}(\mu_{\bar{N}_x}, \sigma_{\bar{N}_x}) \quad (11)$$

$$\mu_{\hat{N}_x} = \hat{N}_x^{explorable} \quad (12)$$

$$\sigma_{\hat{N}_x} = K_{sd} * \overline{ED}_{min} \quad (13)$$

where $P(Z_x)$ is the probability of a random variable Z at $Node_x^{explorable}$. $\hat{N}_x^{explorable}$ is the estimated value from Equation (1) of $Node_x^{explorable}$. K_{sd} is a constant parameter, and \overline{ED}_{min} is the smallest Euclidean distance among distances between that node and explored nodes.

As discussed above, the algorithm ensures that the camera at any iteration, always has enough energy to move back to the previous best node N_{exp_min} . Therefore, a new distribution can be generated, which sets values bigger than the minimum (N_{exp_min}) in the original distribution to be the minimum value. Then Equations (11) to (13) are modified with a new random variable Z_x^{new} and its expectations as follows:

$$Z_x^{new} = \begin{cases} Z_x, & \text{if } Z_x < N_{exp_min} \\ N_{exp_min}, & \text{if } Z_x \geq N_{exp_min} \end{cases} \quad (14)$$

$$P(Z_x^{new}) = P(Z_x) = \mathcal{N}(\mu_{\hat{N}_x}, \sigma_{\hat{N}_x}) \quad (15)$$

$$\hat{N}_x^{explorable} = \sum [Z_x^{new} * P(Z_x^{new})] \quad (16)$$

where $\hat{N}_x^{explorable}$ is the stochastic modified estimated value for the node $Node_x^{explorable}$.

Then a new hash map $\hat{M}_{explorable}$ is set up with pairs as $Node_{index}^{explorable} : (P_{index}^{explorable}, \hat{N}_x^{explorable})$. The next target node to be explored is the one that has the smallest modified estimated value in the map:

$$index_{Next} = \{index | \hat{N}_{index_{Next}}^{explorable} = \min(\hat{N}_{index}^{explorable})\} \quad (17)$$

$$p_{Next} = p_{index_{Next}}^{explorable} \quad (18)$$

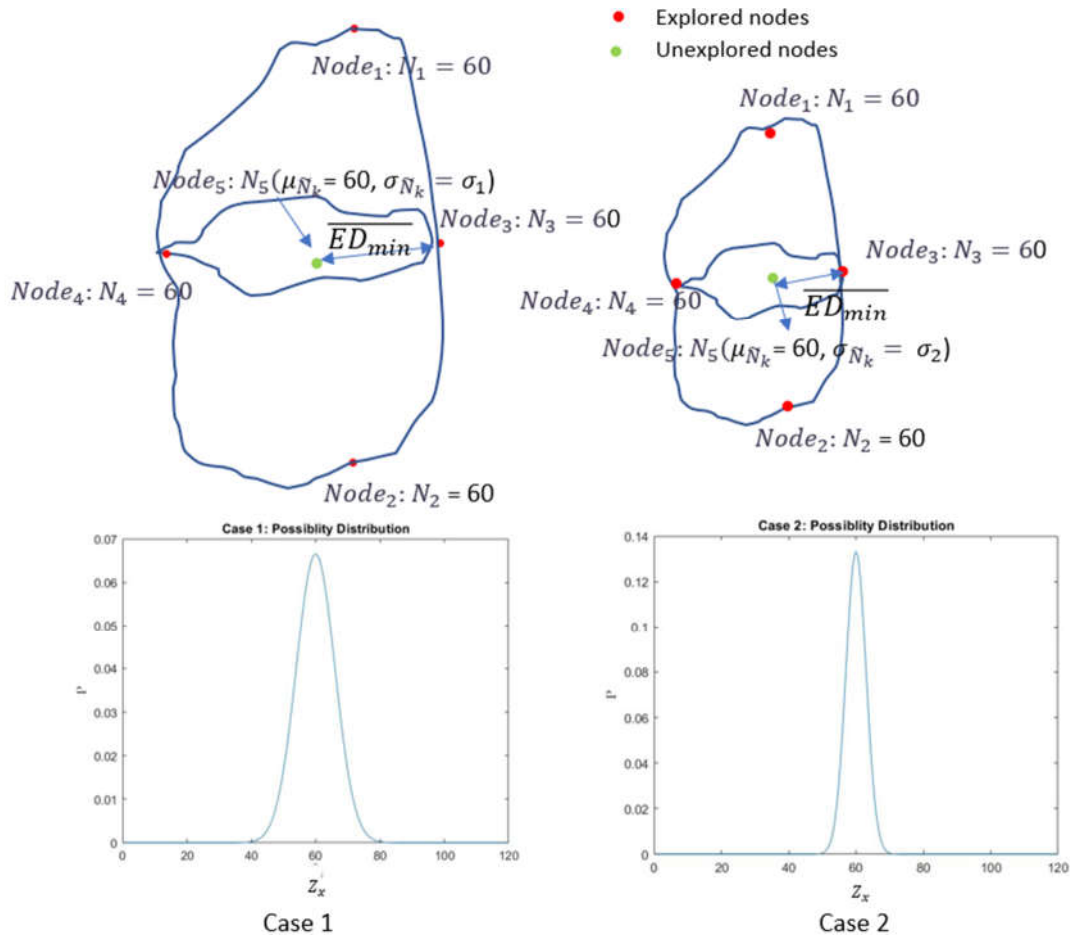


Figure 4. Both case1 and case2 have the same estimated value of an unexplored node (green point) from Equation (1). However, the node in case 1 has a larger distance to explored nodes (red points) compared to that in case 2. Therefore, we set different standard deviations ($\sigma_1 > \sigma_2$) to differentiate possibility distributions. This approach distinguishes nodes that have the same estimations from equation (1) to have different likelihood to be explored in the next iteration.

2.2.4. Estimated Energy Cost Function

As discussed above, an estimated energy cost function $E(P_a, P_b)$ calculates the estimated energy cost from P_a to P_b . This function is used to find feasible node set $S_{feasible}$ and explorable node set $S_{explorable}$. In this section, the equations for the cost function are derived from the robot positioning-controlled system's response. Also, trade-offs between energy cost and settling time of moving are discussed.

The energy of moving the camera between two locations from $P_{initial}$ to P_{final} , results from the energy cost of DC motors in 6 DOFs robot arm's each joint rotating from joint angles $q_{initial}$ to q_{final} . Therefore, $E(P_{initial}, P_{final})$ can be described as the sum of the time integration of rotational power in each joint. That is:

$$E(P_{initial}, P_{final}) = \sum_{i=1}^6 \int_{t_0^i}^{t_f^i} V^i \cdot I^i \cdot dt \quad (19)$$

where V^i is the voltage and I^i is the current in the DC motor's circuit of the i^{th} joint. t_0^i and t_f^i are initial time and final time of the i^{th} joint moving from its initial angle $q_{initial}^i$ to its final angle q_{final}^i .

Without detailed derivation, the dynamic model of a 6dofs revolutionary robotic manipulator and DC motors can be expressed as [20]:

$$\frac{1}{r_k} J_{m_k} \ddot{q}_k + \sum_{j=1}^6 d_{j,k} \ddot{q}_j + \sum_{i,j}^6 c_{i,j,k}(q) \dot{q}_i \dot{q}_j + \frac{1}{r_k} B \dot{q}_k + \Phi_k(q) = \frac{K_m}{r_k R} V^i \quad (20)$$

$$J_m = J_a + J_g, \quad B = B_m + \frac{K_b K_m}{R} \quad (21)$$

$$c_{i,j,k} = \frac{1}{2} \left(\frac{\delta d_{k,j}}{\delta q_i} + \frac{\delta d_{k,i}}{\delta q_j} - \frac{\delta d_{i,j}}{\delta q_k} \right), \quad \Phi_k = \frac{\delta V}{\delta q_k} \quad (22)$$

$$i, j, k \in (1, 2, 3, 4, 5, 6) \quad (23)$$

where Equations (20) and (21) express the k^{th} joint dynamics equation. $q_i/q_j/q_k$ is the joint revolute variable. J_a and J_g represent the moment of inertia of the motor, and the gear of the model. r_k is gear ratio at k^{th} joint, and B_m is the damping effect of the gear. $d_{i,j}$ represents the entry of inertial matrix of the robot manipulator at i^{th} row and j^{th} column. $c_{i,j,k}$ is Christoffel symbols and for a fixed k , with $c_{i,j,k} = c_{j,i,k}$. And Φ_k is the derivative of potential energy V with respect to k^{th} joint variance. K_m is the torque constant in $N - m/amp$, K_b is the back emf constant, and R is Armature Resistance.

Take u^i as the actuator input to the dynamic system (20) measured at i^{th} joint from a designed controller. Therefore, u^i equals to the right-side of Equation (20). That is:

$$V^i = \frac{rR}{K_m} u^i \quad (24)$$

In addition, without derivation, the expression of current in the Laplace domain [20] is:

$$(Ls + R)I^i(s) = V^i(s) - \frac{K_b}{r} s q^i(s) \quad (25)$$

$$\text{Then: } I^i(s) = \frac{1}{(Ls+R)} V^i(s) - \frac{K_b}{r} \frac{s}{(Ls+R)} q^i(s) \quad (26)$$

$$\text{Take the inverse Laplace: } I^i(t) = \frac{1}{L} e^{-\frac{R}{L}t} * V^i(t) - \frac{K_b}{r} \left(\frac{1}{L} - \frac{R}{L^2} \right) e^{-\frac{R}{L}t} * q^i(t) \quad (27)$$

where L is armature inductance, K_b is back emf constant, and $*$ is the convolutional multiplication. Therefore, the instant current in the time domain $I^i(t)$ is a function of the instant voltage $V^i(t)$ and the instant angle $q(t)$.

Various controller designs, such as PID controller [21], and Youla controller [22], of six DoFs revolute robotic manipulators have been well developed in many papers. In this paper, we use a previous Youla controller design [22] with feedback linearization (Figure 5) as the position controller of the robot manipulator that holds the camera.

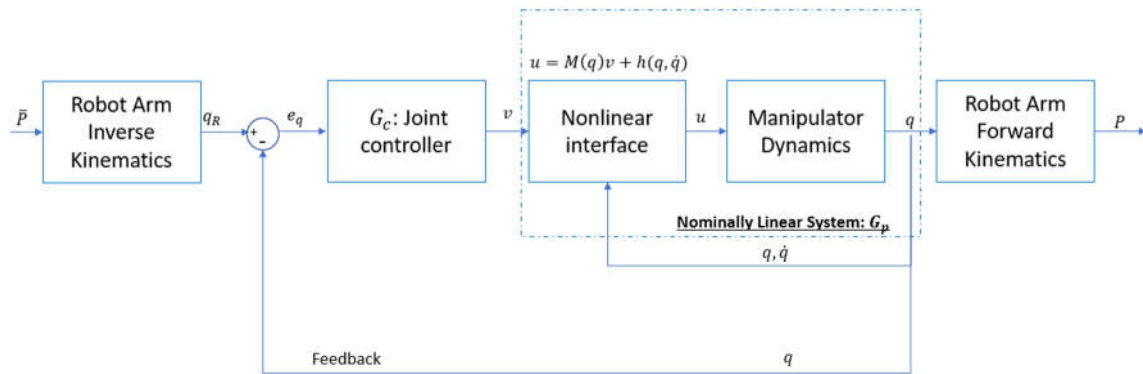


Figure 5. The block Diagram of feedback linearization Youla control design used for the joint control loop.

From Figure 5, the actuator input u^i can be derived from a virtual input v^i so that:

$$u^i = M(q^i(t))v^i + H(q^i(t), \dot{q}^i(t)) \quad (28)$$

where M , and H are nonlinear functions of $q^i(t)$, and $\dot{q}^i(t)$, the first derivative of $q^i(t)$.

Without showing the controller development in this paper, the 1 DoF controller transfer function $G_c(s)$ and 1 DoF nominal plant $G_p(s)$ with v^i as the virtual actuator input in the feedback linearization are expressed as follows:

$$G_c(s) = \frac{3\tau_{in}^2 s + 1}{\tau_{in}^3 s + 3\tau_{in}^2} \quad (29)$$

$$G_p(s) = \frac{1}{s^2} \quad (30)$$

where τ_{in} is a constant parameter in the controller design.

Then the following transfer functions can be calculated as:

$$\frac{v^i(s)}{q_{final}^i(s)} = \frac{G_c}{1 + G_c G_p} \quad (31)$$

$$\frac{q^i(s)}{q_{final}^i(s)} = \frac{G_c G_p}{1 + G_c G_p} \quad (32)$$

$$\frac{\dot{q}^i(s)}{q_{final}^i(s)} = \frac{s G_c G_p}{1 + G_c G_p} \quad (33)$$

By taking inverse Laplace transform, and q_{final}^i as a step input, the following equations in the time domain are:

$$v^i(t) = (q^i_{final} - q^i_{initial}) \left(\frac{1}{\tau_{in}^4} t^2 e^{-\frac{t}{\tau_{in}}} + \frac{-5}{\tau_{in}^3} t e^{-\frac{t}{\tau_{in}}} + \frac{3}{\tau_{in}^2} e^{-\frac{t}{\tau_{in}}} \right) \quad (34)$$

$$q^i(t) = (q^i_{final} - q^i_{initial}) \left(\frac{1}{\tau_{in}^2} t^2 e^{-\frac{t}{\tau_{in}}} - \frac{1}{\tau_{in}} t e^{-\frac{t}{\tau_{in}}} - e^{-\frac{t}{\tau_{in}}} + 1 \right) + q^i_{initial} \quad (35)$$

$$\dot{q}^i(t) = (q^i_{final} - q^i_{initial}) \left(\frac{3}{\tau_{in}^2} t e^{-\frac{t}{\tau_{in}}} - \frac{1}{\tau_{in}^3} t^2 e^{-\frac{t}{\tau_{in}}} \right) \quad (36)$$

With expressions of $v^i(t)$, $q^i(t)$, and $\dot{q}^i(t)$, Equations (24) and (28) shows that:

$$V^i(t) = F(q^i_{initial}, q^i_{final}, t) \quad (37)$$

And combine Equations (27) and (37):

$$I^i(t) = G(q^i_{initial}, q^i_{final}, t) \quad (38)$$

where F, and G are nonlinear functions of $q^i_{initial}$, q^i_{final} , and t .

It has been shown so far that the estimated energy cost $E(P_{initial}, P_{final})$ from location $P_{initial}$ to P_{final} is a function of $q_{initial}$ and q_{final} .

$q_{initial}$ and q_{final} are derived from the inverse kinematics process [20] (Shown in Appendix Equations (B6) -(B13)), that is:

$$q_{initial} = \text{inverkinematics}(P_{initial}) \quad (39)$$

$$q_{final} = \text{inverkinematics}(P_{final}) \quad (40)$$

$$\text{And } q_{initial} = [q^i_{initial} | i \in (1,2,3,4,5,6)], \quad q_{final} = [q^i_{final} | i \in (1,2,3,4,5,6)] \quad (41)$$

Development of Equations (34) to (36) assumes that the target angle of each joint q^i_{final} is a step input. A more realistic assumption is to set q^i_{final} as a delayed input.

$$q^i_{final}(t) = (1 - e^{-\frac{t}{\tau_{delay}}}) q^i_{final} \quad (42)$$

where τ_{delay} is a time constant that measures the time delay of the target in the real positioning control. $\tau_{delay} \geq 0$, and when $\tau_{delay} = 0$, it indicates no delay exists in the input.

The Laplace form of (42) is:

$$q^i_{final}(s) = \left(\frac{1}{s} - \frac{1}{s + \frac{1}{\tau_{delay}}} \right) q^i_{final} \quad (43)$$

With new expression of $q^i_{final}(s)$, Equation (34) to (36) can be developed as:

$$v^i(t) = (q^i_{final} - q^i_{initial}) \left(\frac{A_v}{\tau_{in}^3} t^2 e^{-\frac{t}{\tau_{in}}} + B_v \frac{B_v}{\tau_{in}^2} t e^{-\frac{t}{\tau_{in}}} + \frac{C_v}{\tau_{in}} e^{-\frac{t}{\tau_{in}}} + \frac{D_v}{\tau_{delay}} e^{-\frac{t}{\tau_{delay}}} \right) \quad (44)$$

$$\begin{aligned} A_v &= \frac{1}{\tau_{in}} - \frac{-\tau_{delay}}{\tau_{in}(\tau_{in}-\tau_{delay})} & B_v &= \frac{-5}{\tau_{in}} - \frac{7\tau_{in}\tau_{delay}-5\tau_{delay}^2}{\tau_{in}(\tau_{in}-\tau_{delay})^2} \\ C_v &= \frac{3}{\tau_{in}} - \frac{\tau_{delay}(-8\tau_{in}^2+9\tau_{in}\tau_{delay}-3\tau_{delay}^2)}{\tau_{in}(\tau_{in}-\tau_{delay})^3} & D_v &= \\ & & & - \frac{3\tau_{in}\tau_{delay}-\tau_{delay}^2}{(\tau_{in}-\tau_{delay})^3} \end{aligned}$$

$$q^i(t) = (q^i_{final} - q^i_{initial}) \left(\frac{A_Q}{\tau_{in}^3} t^2 e^{-\frac{t}{\tau_{in}}} + \frac{B_Q}{\tau_{in}^2} t e^{-\frac{t}{\tau_{in}}} + \frac{C_Q}{\tau_{in}} e^{-\frac{t}{\tau_{in}}} + \frac{D_Q}{\tau_{delay}} e^{-\frac{t}{\tau_{delay}}} + 1 \right) + q^i_{initial} \quad (45)$$

$$\begin{aligned} A_Q &= \tau_{in} + \frac{\tau_{in}\tau_{delay}}{(\tau_{in}-\tau_{delay})} & B_Q &= -\tau_{in} - \frac{3\tau_{in}^2\tau_{delay}-\tau_{in}\tau_{delay}^2}{(\tau_{in}-\tau_{delay})^2} \\ C_Q &= -\tau_{in} - \frac{-3\tau_{in}^2\tau_{delay}^2+\tau_{in}\tau_{delay}^3}{(\tau_{in}-\tau_{delay})^3} & D_Q &= -\frac{3\tau_{in}\tau_{delay}^3-\tau_{delay}^4}{(\tau_{in}-\tau_{delay})^3} \end{aligned}$$

$$\dot{q}^i(t) = (q_{final}^i - q_{initial}^i) \left(\frac{A_{\dot{Q}}}{\tau_{in}^4} t^2 e^{-\frac{t}{\tau_{in}}} + \frac{B_{\dot{Q}}}{\tau_{in}^3} t e^{-\frac{t}{\tau_{in}}} + \frac{C_{\dot{Q}}}{\tau_{in}^2} e^{-\frac{t}{\tau_{in}}} + \frac{D_{\dot{Q}}}{\tau_{delay}^2} e^{-\frac{t}{\tau_{delay}}} \right) \quad (46)$$

$$A_{\dot{Q}} = -A_Q \quad B_{\dot{Q}} = 2A_Q - B_Q$$

$$C_{\dot{Q}} = B_Q - C_Q \quad D_{\dot{Q}} = -D_Q$$

A simulation scenario is set up to calculate how estimated energy cost changes with varying τ_{delay} . Set $P_{initial} = [-0.30, 0.05, 1.20]$, and $P_{final} = [-0.45, 0.45, 1.20]$ and use the equations above. Figure 6 shows the response of one angle $q^1(t)$ with varying τ_{delay} and Table 1 presents the estimated energy cost: $E(P_{initial}, P_{final})$ and settling time: t_s with varying τ_{delay} .

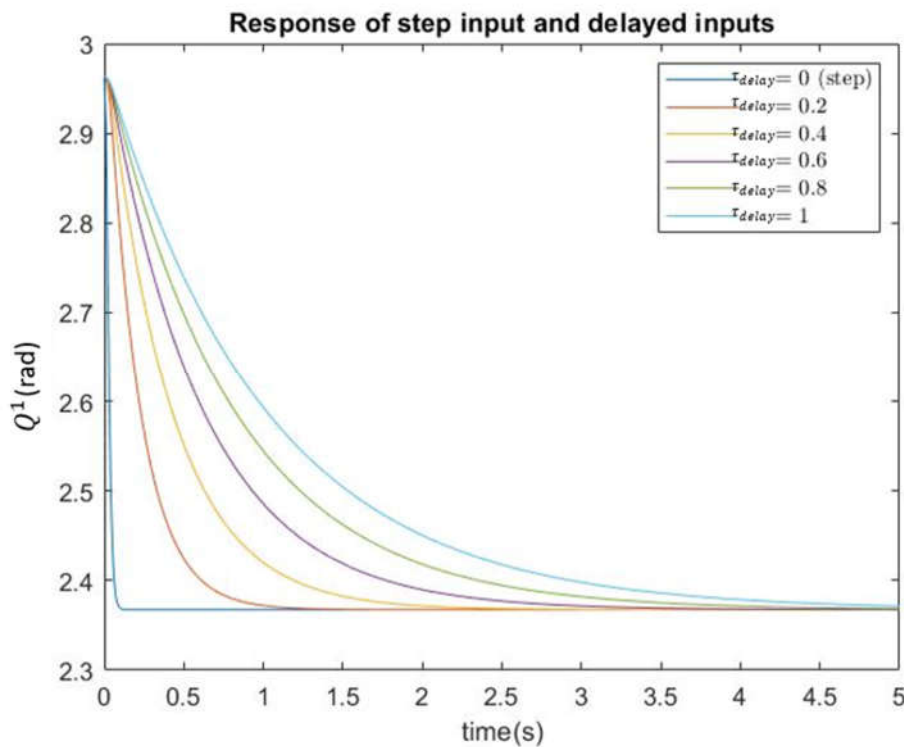


Figure 6. Response of $q^1(t)$ with varying τ_{delay} .

Table 1. Estimated Energy Cost and Settling Time with Varying Delay Constant.

τ_{delay} [s]	$E(P_{initial}, P_{final})$ [ws]	t_s [s]
0	10.895	0.07
0.2	0.102	0.73
0.4	0.030	1.42
0.6	0.016	2.11
0.8	0.011	2.80
1	0.009	3.49

Note: $E(P_{initial}, P_{final})$ is measured as [watts seconds]

Table 1 clearly shows a tradeoff between the estimated energy cost and settling time; reducing the energy cost of moving the camera inevitably increases the response time. This finding matches the results in Figure 6.

If the delay of input is given or can be measured, the estimated energy cost can be calculated through the process in this section. However, if the value is unknown, the delay constant must be chosen and decided based on the following criteria:

- Select small τ_{delay} for a conservative algorithm that searches a small area but ensures it ends up with the minimum that has been explored.
- Select large τ_{delay} for an aggressive algorithm that searches a large area but risks not ending up at the minimum that has been explored.

In this section, the estimated energy cost function is well developed. The parameter values of motors and gears used in the simulations are summarized in Appendix Table A4.

3. Results and Discussion

The developed algorithm has been simulated on an application in automated manufacturing (Figure 7). This is a multi-robot system composed of a visual system and a tool manipulation system. In the visual system, a camera is mounted on an elbow robot arm while a tool is held by the end-effector of the robot manipulator arm. The goal of the visual system is to provide a precise estimation of the tool pose so that the tool manipulator can control the pose with guidance from the visual system. The first step of the manufacturing process is to move the camera in the space and search for the best location so that the number of images required for averaging is minimized to reduce the image noises within an acceptable limit. The model of both robot manipulators is ABB IRB 4600 [23] and the model of the stereo camera we choose in this project is Zed 2 [24].

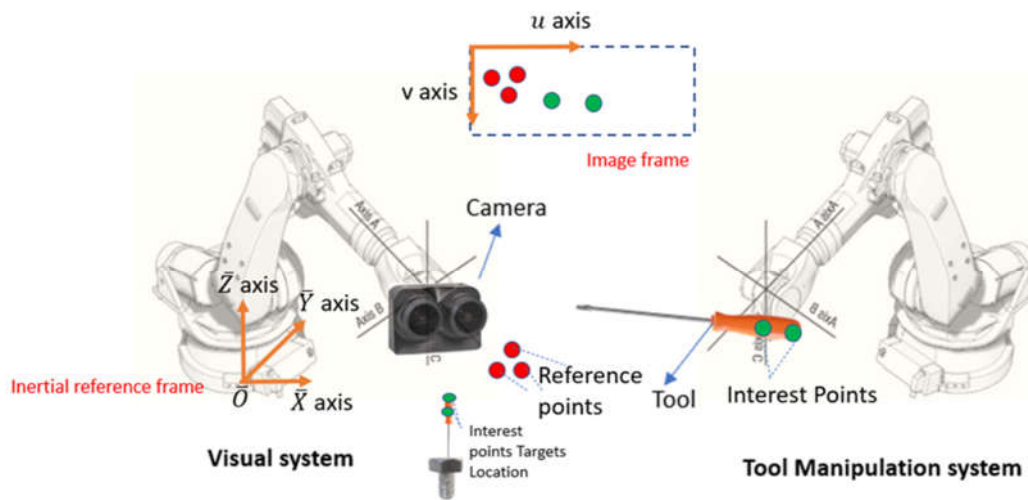


Figure 7. The topology of the multi-robotic system for accurate positioning control.

In this section, we first define the camera's operational space in this application and then provide the simulation results of a made-up scenario.

3.1. Camera Operational Space

In the above discussion, a camera operational space is gridded with nodes and each node is a potential candidate of the camera's optimal position. The camera operational space is defined by three geometric constraints below:

1. The camera can only be allowed to locations where fiducial markers that are attached to the tool are recognizable on the image frame. Therefore, the fiducial markers are in the angle of view of the camera, and the distance between the markers and the camera center is within a threshold.
2. The camera can only be allowed to locations within the reach of the robot arm.

3. The camera can only be allowed in locations where the visual system and the tool manipulation system do not physically interfere with each other.

From the specifications of the stereo camera and the robot arm (Appendix Table A1 and A3), the geometry and dimensions of the camera operational space are analyzed in the following part of the subsections.

3.1.1. Reachable and Dexterous Workspace of Two-Hybrid Systems

In the spherical wrist model of the robot arm, three rotational axes represent the pitch, roll, and yaw of the end-effector independently. As shown in Figure 8, those three axes intersect at point H . Point P is the location of the end-effector and it's the place where the camera or tool is mounted. For the elbow manipulator mounted with the stereo camera, the camera is placed so that its optical center line coincides with axis C. The location of camera optical center C is determined and a good estimation of it can be found in this paper [25]. For now, assume the optical center is in the middle of \overline{PJ} (set $\overline{PC} = 17\text{ mm}$). For the elbow manipulator mounted with the tool, the screwdriver is attached at the point P . Point J indicates the far end of the object (tool or camera) attached to the end effector. Point H is kept stationary no matter which axis rotates. Therefore, the task of positioning and rotation is decoupled in the spherical wrist robot model.

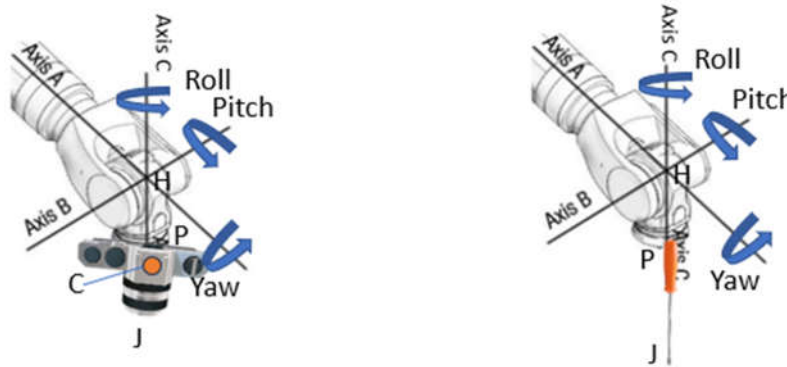


Figure 8. Spherical Wrist Model.

The idea of reachable and dexterous workspaces of an ideal elbow manipulator has been introduced in the paper [26]. An ideal elbow manipulator is a manipulator whose angles of rotation are free to move in the whole operational range $[0, 2\pi]$. However, a realistic elbow manipulator is limited to moving its joints within certain ranges of angle.

The workspace of an ideal elbow manipulator (Figure 9) is a sphere centered at the joint 1 of the manipulator, denoted as point o . The reachable workspace concerning the center o of a manipulator is the aggregate of all possible locations of the point J attached with the end-effector and is denoted as $W_o(J)$. The dexterous workspace to the center o of a manipulator is the aggregate of all possible locations that point J can reach all possible orientations of the end-effector and is denoted as $W_o^d(J)$. The reachable workspace of H is denoted as $W_o(H)$.

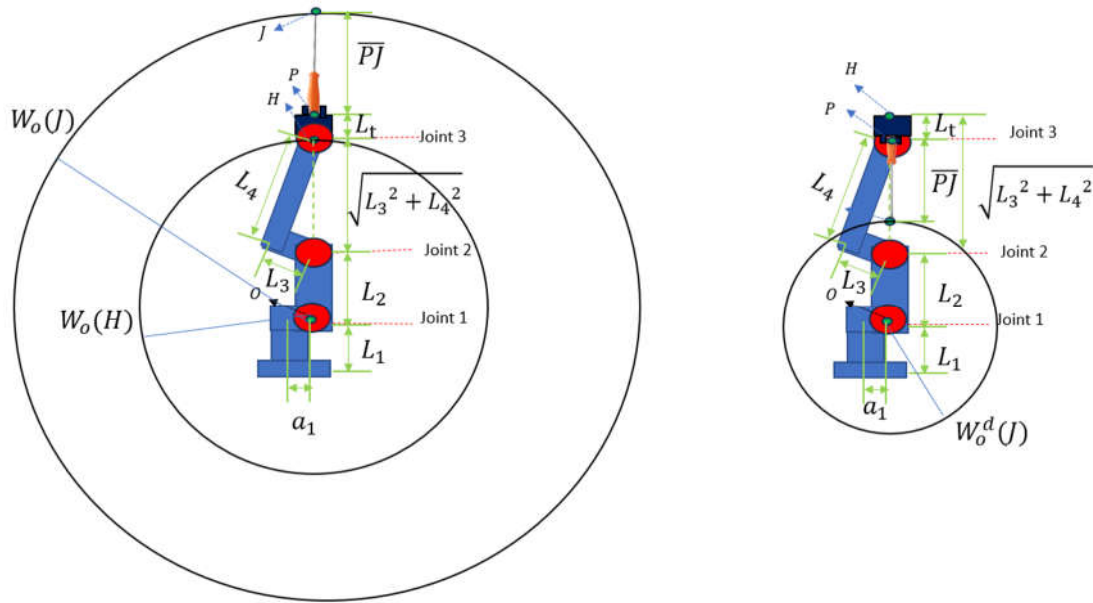


Figure 9. Workspace of an Ideal Elbow Manipulator.

It is clearly shown in the figure:

$$W_o^d(J) \subseteq W_o(H) \subseteq W_o(J). \quad (47)$$

For an ideal elbow manipulator, the radii of workspaces are expressed as below:

$$\text{Radius of } W_o(H) = L_2 + \sqrt{L_3^2 + L_4^2} \quad (48)$$

$$\text{Radius of } W_o(J) = L_2 + \sqrt{L_3^2 + L_4^2} + L_t + \overline{PJ} \quad (49)$$

$$\text{Radius of } W_o^d(J) = L_2 + \sqrt{L_3^2 + L_4^2} - L_t - \overline{PJ} \quad (50)$$

where L_2 , L_3 , and L_4 are links' length of the manipulator. L_t is the length of the end-effector and \overline{PJ} is the length of the object that mounts on the end-effector.

Because the camera and the tool must be able to rotate in all 3 DoFs when they are at the target position. So, the dexterous workspace is used as the working space of the visual system and the tool manipulation system. Dexterous workspace of the optical center is used for the visual system and Dexterous workspace of the far endpoint of the tool is used for the tool manipulation system.

$$\begin{aligned} \text{Radius of } W_o^d(\text{Visual}) &= r_V = L_2 + \sqrt{L_3^2 + L_4^2} - L_t - \overline{PC}_{\text{camera}} \\ &= 1.716 \text{ m (From Specification Appendix Table A1)} \end{aligned} \quad (51)$$

$$\begin{aligned} \text{Radius of } W_o^d(\text{Tool}) &= r_T = L_2 + \sqrt{L_3^2 + L_4^2} - L_t - \overline{PJ}_{\text{tool}} \\ &= 1.606 \text{ m (From Specification Appendix Table A1)} \end{aligned} \quad (52)$$

Figure 9 shows the workspace setup for the hybrid system. The dexterous space of robot arm with camera is a sphere with radius of r_V and sphere center is O_V and the dexterous space of the robot arm with tools is a sphere the radius is r_T and center of this sphere is O_T . Two systems intersect with the ground with an angle θ_1 and θ_2 .

$$\theta_1 = \arcsin\left(\frac{L_1}{r_V}\right) = 16.77^\circ \quad (53)$$

$$\theta_2 = \arcsin\left(\frac{L_1}{r_T}\right) = 17.95^\circ \quad (54)$$

To avoid interference of the visual and the tool manipulation systems (as the third geometric constraint of the operational space), the reachable workspaces (the maximum reach) of two systems

should have no overlap. The reachable workspaces of two systems are calculated from Equation (49) is:

$$\begin{aligned} \text{Radius of } W_o(\text{Visual}) = r_{Vr} &= L_2 + \sqrt{L_3^2 + L_4^2} + L_t + \overline{PJ}_{\text{camera}} \\ &= 2.044 \text{ m (From Specification Appendix Table A1)} \end{aligned} \quad (55)$$

$$\begin{aligned} \text{Radius of } W_o(\text{Tool}) = r_{Tr} &= L_2 + \sqrt{L_3^2 + L_4^2} + L_t + \overline{PJ}_{\text{tool}} \\ &= 2.138 \text{ m (From Specification Appendix Table A1)} \end{aligned} \quad (56)$$

Let L_{VT} is the distance between O_V and O_T . The following relationship must be satisfied:

$$L_{VT} \geq r_{Vr} + r_{Tr} = 4.182 \text{ m} \quad (57)$$

The visual system is able to detect the tool when it gets close to the target pose (Just above the bolt). Also reference points are put near the tool's target pose. Assume a marker is put at where the bolt is, and this marker represents as an approximated location of all interest points and reference points. Therefore, the marker should be within the dexterous space of the tool manipulation system. Set up a coordinate system with \bar{O}_V (the projection of O_V on the ground with a_1 offset) as the origin. The projection of O_T on the ground with a_1 offset is \bar{O}_T . Assume the marker is placed on the line $\bar{O}_V\bar{O}_T$. The distance m from marker M to \bar{O}_V should be:

$$L_M \geq a_1 + L_{VT} - r_T \cos(\theta_2) \geq 2.829 \text{ m} \quad (58)$$

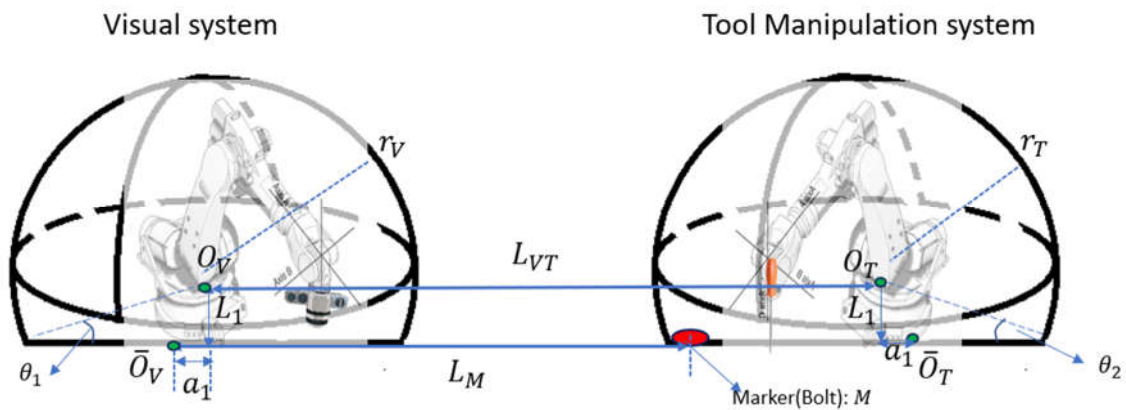


Figure 9. The workspace configuration of the multi-robotic system.

3.1.2. Detectable Space for the Stereo Camera

From Appendix Table A3, the angle of view in width and height for both lens in stereo camera is $\alpha = 86.05^\circ$, and $\beta = 55.35^\circ$. The detectable space for each lens can be modeled as the inner area of a cone with angle α and β . And the overlapped space is the detectable space for the stereo camera. The model is shown below (Figure 10). Point C is the center of the baseline whose length is denoted as b or center of the camera system. The overlap area is also a cone with angles of vertex α and β with the offset d from baseline. Point Q is the vertex of the cone.

$$d = \frac{b/2}{\tan(\alpha/2)} = 0.064 \text{ m} \quad (59)$$

There is an upper bound of the distance between the object to the camera center; if the object is too far away from the camera center, the dimensions of projected images are too small to be measured. Suppose to have a clear image of the fiducial markers (circle shapes), it is required that the diameter of the projected image should takes at least 5-pixel numbers in the image frame:

$$N_H \geq 5 \text{ resolution} \quad (60)$$

where N_H is the pixel number in the image frame.

Select the Zed camera's mode so that its resolution is 1920*1082 and the image sensor size is 5.23mm X 2.94mm. Then numbers resolution in unit length is:

$$n = 367 \text{ resolution/mm} \quad (61)$$

Then the range of markers' diameter on the image frame is:

$$d_H = \frac{N_H}{n} \geq 0.0136 \text{ mm} \quad (62)$$

Also assume in the inertial frame, the diameter of attached fiducial marks is:

$$D_H = 12 \text{ mm} \quad (63)$$

Then from pinhole model of the camera, the range of distance between markers and camera center Z is:

$$Z = \frac{f_u D_H}{d_H} \leq 2.47 \text{ m} = Z_{max} \quad (64)$$

where Z_{max} is the maximum depth of camera to detect the markers. This parameter defines the height of the cone in Figure 10. The detectable space (abiding the second geometric constraint) forms an elliptic cone with different angles of vertex.

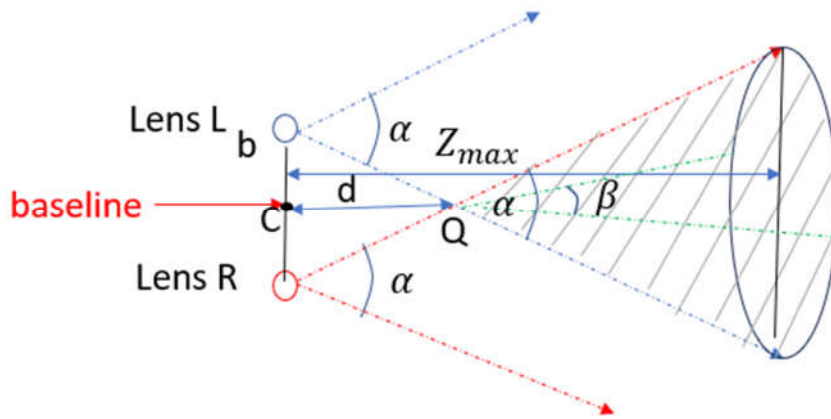


Figure 10. The Detectable Space of the Stereo Camera.

3.1.3. Camera Operational Space Development

In Figure 11, set the camera lens so that it is always parallel to the face of the marker and the baseline is parallel to \bar{X}_V axis. In this case, lens is kept facing downward when detecting the marker on the ground. As discussed in the above section, the detectable space of the camera is modeled as a cone. This cone intersects with the horizontal plane and forms an oval. To have the camera detect the marker, the marker must be contained inside the oval.

An inertia coordinate system $\bar{O}_V \bar{X}_V \bar{Y}_V \bar{Z}_V$ is set up with its origin seated at the bottom center of the visual system. Make point C as the location of the camera center and its coordinates in the inertial frame are (X_C^V, Y_C^V, Z_C^V) . Q is the vertex of the detectable elliptic cone, and Q' is the projection of Q on the horizontal plane. M is the marker.

The elliptic cone of the camera's detectable space always intersects with the horizontal plane. Consider an extreme case when point C is located at the apex of the visual system workspace as shown in Figure 11. Then the Z_C^V coordinate of C at the inertial frame is:

$$Z_C^V \{max\} = L_1 + r_v = 2.211 \text{ m} \leq Z_{max} \quad (65)$$

where Z_{max} defined above is the maximum depth of camera to detect the markers. Therefore, the cone intersects with the horizontal plane when the camera center C is placed at any location of the workspace's contour.

From expression of Cartesian coordinates in the inertia frame:

$$a_{Q'} = (Z_C^V - d)\tan(\frac{\alpha}{2}), \quad b_{Q'} = (Z_C^V - d)\tan(\frac{\beta}{2}) \quad (66)$$

where d is the same offset in Equation (59). $a_{Q'}$ and $b_{Q'}$ are major radius and minor radius of the oval. Also coordinates of points Q' and M are $(X_C^V, Y_C^V, 0)$ and $(L_M, 0, 0)$. Q' is the center of the oval. To ensure the marker M is inside the projected oval, it is required that:

$$\frac{(X_C^V - L_M)^2}{a_{Q'}^2} + \frac{(Y_C^V)^2}{b_{Q'}^2} \leq 1 \quad (67)$$

Plug Equation (66) into Equation (67):

$$\frac{(X_C^V - L_M)^2}{\tan(\frac{\alpha}{2})^2} + \frac{(Y_C^V)^2}{\tan(\frac{\beta}{2})^2} \leq (Z_C^V - d)^2 \quad (68)$$

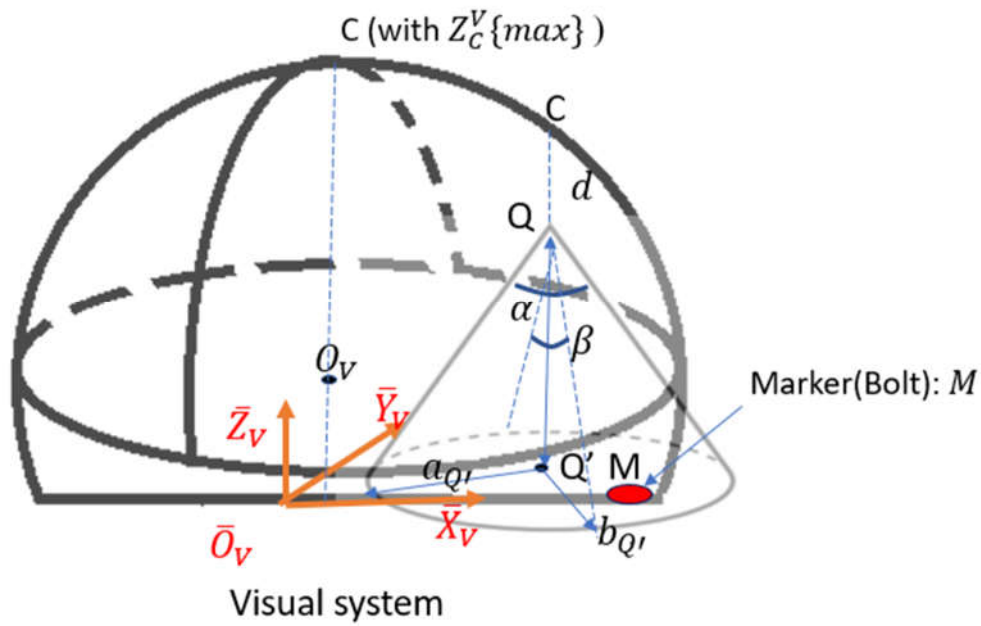


Figure 11. The Detectable Elliptic Cone in the Inertial Visual System Frame.

Inequality (68) is exactly a mathematic expression of all points $C(X_C^V, Y_C^V, Z_C^V)$ within a cone whose opening is in the positive Z direction with its vertex at $(L_M, 0, d)$ and the opening parameters as $\tan(\frac{\alpha}{2})=0.934$ and $\tan(\frac{\beta}{2})=0.524$. Therefore, this inequality defines the second geometric constraint of the camera center. It forms an elliptical cone with its vertex located in Figure 12 at point E, which is the offset point of the marker M from the ground. As $C(X_C^V, Y_C^V, Z_C^V)$ should also be within the sphere of the workspace (abiding the first geometric constraint), the camera operational space is presented as the overlap between the cone and the sphere (the shaded area in Figure 12).

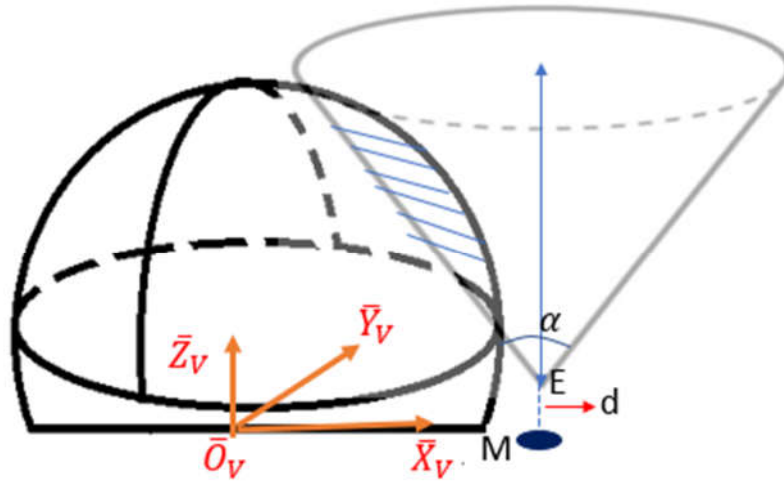


Figure 12. Illustration of Camera Operational Space (the shaded area).

From Figure 12, the marker should not be too far from the visual system, otherwise there is no overlap region between the cone and the sphere. The largest value of L_M , which is the distance from the marker to the coordinate center, occurs when the cone is tangent to the sphere as shown in Figure 13. To find the limit of L_M , draw a horizontal line across point O_V so that it intersects the surface of the cone at point A and the center line of the cone at point B as shown in Figure 13. Let $\overline{O_V A}$ and \overline{AB} are line segments' length.

Therefore, from the geometry, the upper limit of L_M is:

$$L_M < \overline{O_V A} + \overline{AB} + a_1 = \frac{r_V}{\cos\left(\frac{\alpha}{2}\right)} + (L_1 - d) \tan\left(\frac{\alpha}{2}\right) + a_1 \quad (69)$$

Combining with the lower bound of L_M in Equation (58), the marker M should be placed in the system so that:

$$2.829 \text{ m} \leq L_{VT} - r_T \cos(\theta_2) \leq L_M < \frac{r_V}{\cos\left(\frac{\alpha}{2}\right)} + (L_1 - d) \tan\left(\frac{\alpha}{2}\right) + a_1 = 2.925 \text{ m} \quad (70)$$

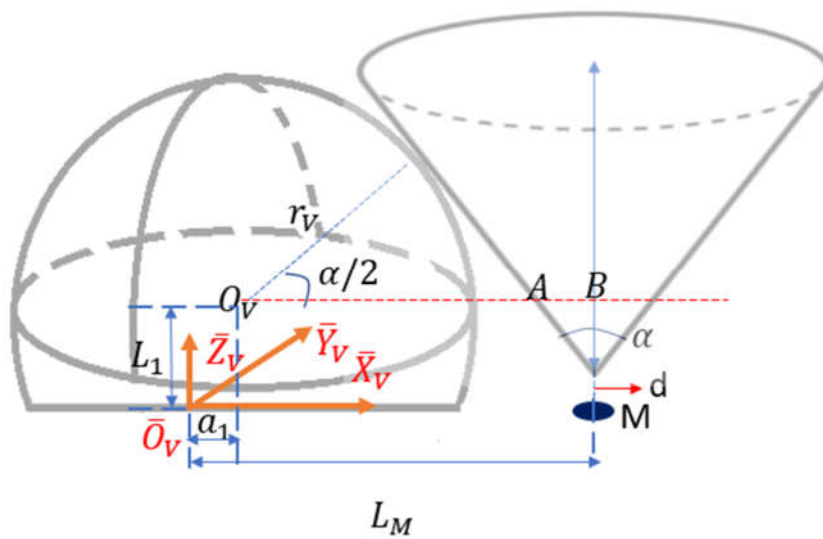


Figure 13. An Extreme Case of Marker Position.

3.1.4. Mathematic Expression for Node Coordinates Within the Camera Operational Space

From Equation (70), the closer the marker's distance L_M to its lower bound, the larger the operational space is. In general, larger operational space is preferred, because the camera has a larger space for searching the optimal location. Set L_M equals to its lower bound:

$$L_M = 2.83m \quad (71)$$

Get the cross-section area at $\bar{X}_V - \bar{Z}_V$ plane (Figure 14).

The function of semi-circle and the line EFG is:

$$(\bar{Z}_V - L_1)^2 + (\bar{X}_V - a_1)^2 = r_V^2 \quad (72)$$

$$\bar{X}_V = -\tan\left(\frac{\alpha}{2}\right)(\bar{Z}_V - d) + L_M \quad (73)$$

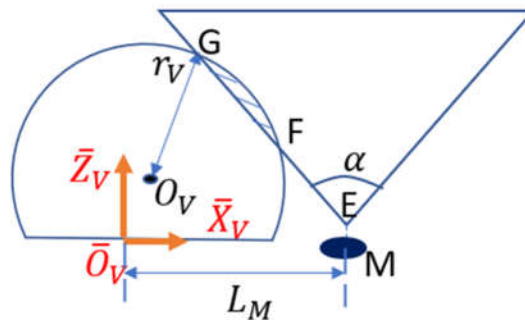


Figure 14. The Camera Operational Space Cross-section at $\bar{X}_V - \bar{Z}_V$ plane.

Combine Equations (72) and (73) and solve:

$$(1 + \tan^2\left(\frac{\alpha}{2}\right))\bar{Z}_V^2 - 2(\tan\left(\frac{\alpha}{2}\right)^2 d + \tan\left(\frac{\alpha}{2}\right)(L_M - a_1) + L_1)\bar{Z}_V + \tan\left(\frac{\alpha}{2}\right)^2 d^2 + 2 \tan\left(\frac{\alpha}{2}\right)(L_M - a_1)d + (L_M - a_1)^2 - r_V^2 + L_1^2 = 0 \quad (74)$$

Solve Equation (74) by plugging numbers:

$$1.872 \bar{Z}_V^2 - 6.061 \bar{Z}_V + 4.670 = 0 \quad (75)$$

$$\text{Solve: } \bar{Z}_{V_{max}}, \bar{Z}_{V_{min}} = 1.972 m, 1.265 m \quad (76)$$

$\bar{Z}_{V_{max}}$ and $\bar{Z}_{V_{min}}$ are the largest and smallest \bar{Z}_V coordinates of the camera center C that is within the camera operation space.

Take a value of Z_C^V (\bar{Z}_V coordinates of the camera center C) in the range $(\bar{Z}_{V_{min}}, \bar{Z}_{V_{max}}) = (1.265 m, 1.972 m)$. Set a plane $\bar{Z}_V = Z_C^V$, and that plane intersects the camera operational space and forms a shade area, which is the overlap between the circle and the oval, as shown in Figure 14. The inequality governing points C within sphere at a specific Z_C^V is:

$$(\bar{X}_V - a_1)^2 + \bar{Y}_V^2 \leq r_V^2 - (Z_C^V - L_1)^2 \quad (77)$$

The inequality governing points C within cone at Z_C^V is:

$$\frac{(\bar{X}_V - L_M)^2}{\tan^2\left(\frac{\alpha}{2}\right)} + \frac{(\bar{Y}_V)^2}{\tan^2\left(\frac{\beta}{2}\right)} \leq (Z_C^V - d)^2 \quad (78)$$

By plotting those inequalities in Figure 15, the shaded area is where X_C^V and Y_C^V (\bar{X}_V and \bar{Y}_V coordinates of the camera center C) should be located within. The intersection of the two curves occurs when equality holds for (77) and (78), that is:

$$(\bar{X}_V - a_1)^2 + \bar{Y}_V^2 = r_V^2 - (Z_C^V - L_1)^2 \quad (79)$$

$$\frac{(\bar{X}_V - L_M)^2}{\tan^2\left(\frac{\alpha}{2}\right)} + \frac{(\bar{Y}_V)^2}{\tan^2\left(\frac{\beta}{2}\right)} = (Z_C^V - d)^2 \quad (80)$$

Solve by plugging numbers:

$$(\bar{Y}_{V_{max}}, \bar{Y}_{V_{min}}) = \pm 0.5 * \sqrt{3.44 * Z_C^V - 2.0241 * Z_C^V - 29.802 + 2.451 * \sqrt{-29.475 * Z_C^V + 23.707 * Z_C^V + 137.28}} \quad (81)$$

$\bar{Y}_{V_{max}}$ and $\bar{Y}_{V_{min}}$ are the largest and the smallest \bar{Y}_V coordinates of the camera center C in the camera operation space with specific Z_C^V .

Take Y_C^V in the range $(\bar{Y}_{V_{min}}, \bar{Y}_{V_{max}})$, then for specific X_C^V :

$$\bar{X}_{V_{max}} = \sqrt{r_V^2 - (Z_C^V - L_1)^2 - Y_C^{V^2} + a_1} \quad (82)$$

$$\bar{X}_{V_{min}} = L_M - \sqrt{(Z_C^V - d)^2 \tan^2\left(\frac{\alpha}{2}\right) - Y_C^{V^2} \frac{\tan^2\left(\frac{\alpha}{2}\right)}{\tan^2\left(\frac{\beta}{2}\right)}} \quad (83)$$

$\bar{X}_{V_{max}}$ and $\bar{X}_{V_{min}}$ are the largest and the smallest \bar{X}_V coordinates of the camera center C in the camera operation space with specific Z_C^V and Y_C^V .

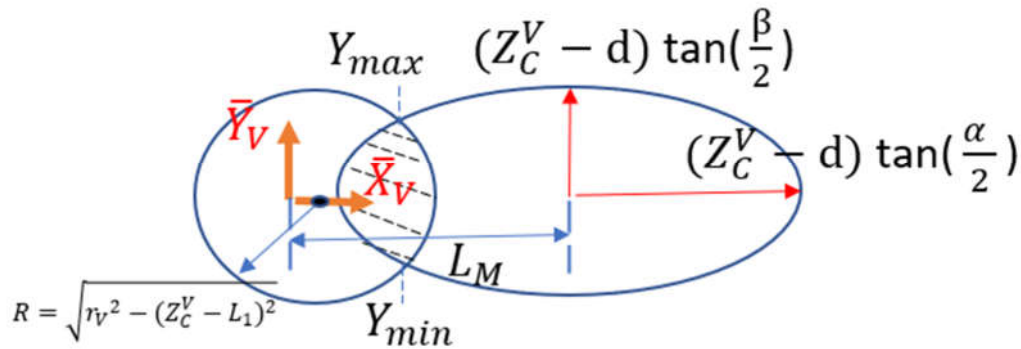


Figure 15. The Camera Operational Space Cross-section at $\bar{X}_V - \bar{Y}_V$ plane.

3.1.5. Numerical Solution of the Ideal Camera Operation Space

With known parameters from the camera and robot arm specifications, all possible location of the camera center $C(X_C^V, Y_C^V, Z_C^V)$ inside the camera operational space can be derived from the following steps:

1. Calculate $\bar{Z}_{V_{max}}$ and $\bar{Z}_{V_{min}}$ by Equation (74), and mesh grid Z_C^V in $(\bar{Z}_{V_{min}}, \bar{Z}_{V_{max}})$.
2. Take a mesh value Z_C^V and calculate $\bar{Y}_{V_{max}}$ and $\bar{Y}_{V_{min}}$ by Equations (79) and (80), and mesh grid Y_C^V in $(\bar{Y}_{V_{min}}, \bar{Y}_{V_{max}})$.
3. Take a mesh Y_C^V and calculate $\bar{X}_{V_{max}}$ and $\bar{X}_{V_{min}}$ by Equations (82) and (83), and mesh grid X_C^V in $(\bar{X}_{V_{min}}, \bar{X}_{V_{max}})$.

Each above computed $C(X_C^V, Y_C^V, Z_C^V)$ is inside the camera operation space. And those three steps completely account for all geometric constraints defined above. The number of mesh points (nodes available for searching) depends on the mesh grid size.

3.1.6. The Camera Operational Space from the Realistic Robot Manipulator

The above procedures of finding the camera operational space assumes the use of ideal robotic manipulators, whose joints are freely to move from $[0, 2\pi]$. However, for realistic robotic manipulators, their revolutionary movement is limited within certain ranges of angles, and the ranges of motion in every joint of the specific robot manipulator: ABB IRB 4600 are listed in Appendix Table A2.

The geometric shape of the operational space for a realistic robotic manipulator is usually irregular (unlike an ideal manipulator whose operational space is the combination of a sphere and a cone) and the mathematical methods of directly finding the nodes within the space is

computationally expensive. A faster way of generating realistic camera operational space is to decrease the ideal operational space by taking out nodes that are out of the operational ranges. Those outlier nodes can be found from the inverse kinematic model of the robotic manipulator. Figure 16 shows the decreased nodes from the ideal camera operational space to the realistic space.

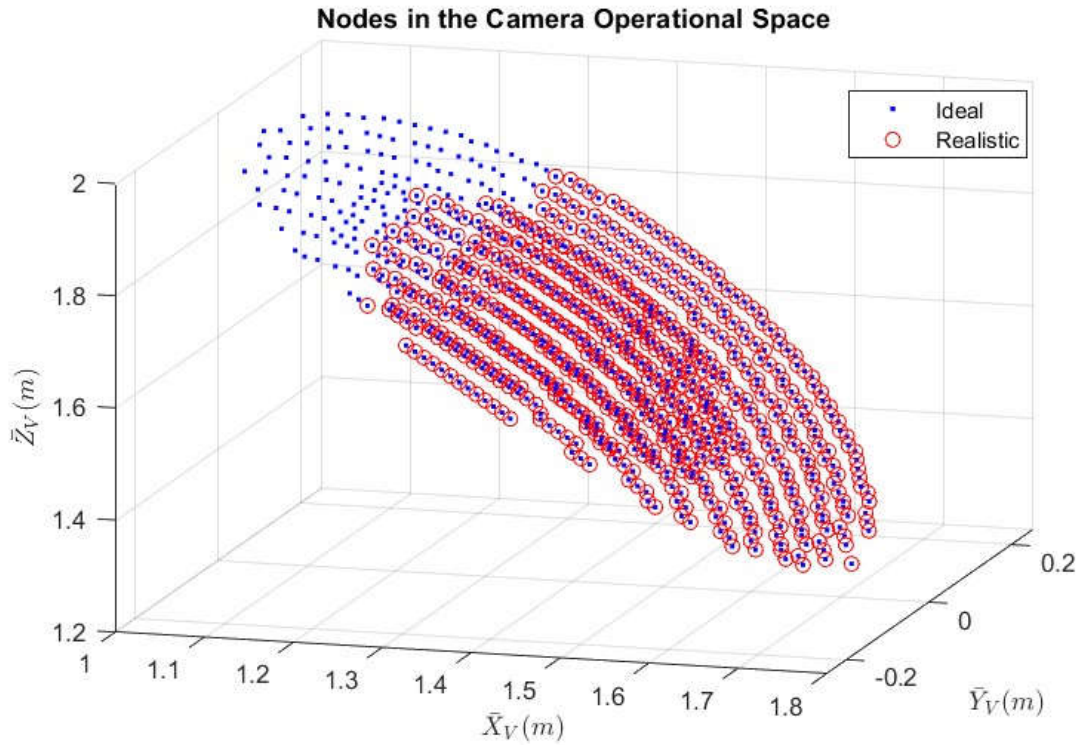


Figure 16. Reduction of Nodes from the ideal to the realistic camera operational space.

3.2. Simulations of scenarios

A simulation in MATLAB is used to validate the proposed algorithm. In the simulation, the environment is set up with two minimums in the space (only one is the global minimum). K_{est} and K_{sd} are tunable parameters and set as $K_{est} = 5$, and $K_{sd} = 50 \text{ (m}^{-1}\text{)}$. The initial available energy bound is set to be $E_{bound} = 12$ (Watt Second (ws) or Joule). Figure 17 shows the sequences of nodes being explored in iteration steps by setting different energy threshold E_T .

By trials of different simulation runs, it can be concluded that:

- When E_T is large (in this scenario, $3ws < E_T \leq 12ws$), the algorithm is conservative, so that it has only searched a small area that excludes the global minimum.
- When E_T is in mid-range (in this scenario, $0.5ws \leq E_T \leq 3ws$), the algorithm drives the camera to the global minimum.
- When E_T is small (in this scenario, $0ws \leq E_T < 0.5ws$), the algorithm is aggressive. Although it has searched a large amount of areas that includes the global minimum, it doesn't have enough energy to move the camera back to the best location explored in the previous stages

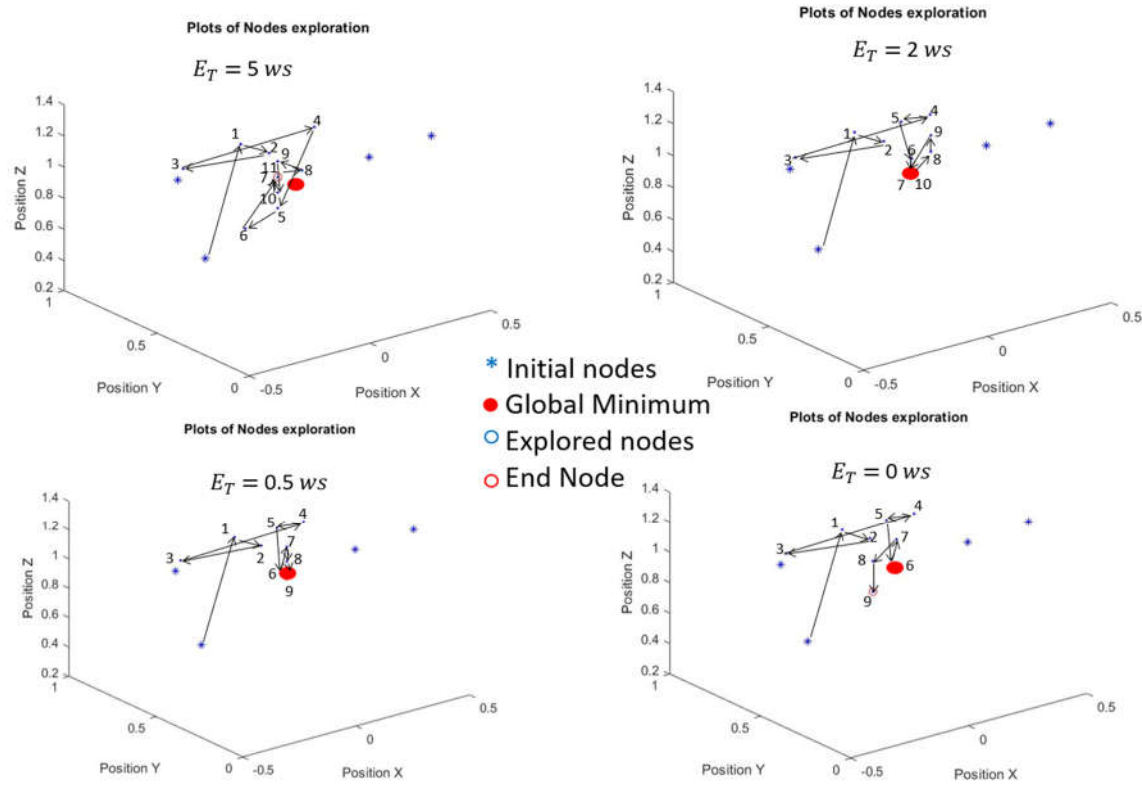


Figure 17. Plots of sequential exploration with different E_T .

The two tunable parameters K_{est} and K_{sd} are selected randomly in the simulation. Those parameters are used in the models to estimate the noise spatial distribution in the environment. Future research can focus on development of adaptive algorithms that tune K_{est} and K_{sd} over iterations based on errors between real measurements and estimations.

In addition, the numerical values of E_T are also picked randomly for testing. In real applications, a fixed E_T should be selected before the operation of this algorithm. The selection of E_T is related to the specific application scenario, size of camera's operational space, and total energy available E_{bound} at beginning. However, the general thumb of the rule is the following: choose small values of E_T when both E_{bound} and operational space is large to encourage exploring over exploiting; otherwise, choose large E_T .

4. Conclusion

In this article, we have developed an algorithm that a camera can explore the workspace in a manufacturing environment and search for a location so that images' noise is minimized within the space that it can reach. This article also provides detailed development of the camera's operational space for a specific application. Simulation results show how energy threshold E_T affects the behavior of the algorithm. A moderate threshold is suggested so that the algorithm can search for enough amount of area but enables to drive the camera back to the minimum with limited energy available. The analysis of E_T 's value in different applications is a focus of the future work. Future work may also include adaptive algorithms that update K_{est} and K_{sd} online. Also, orientations of the camera are fixed in this paper. In the future, we would like to develop advanced algorithms that provide not only the optimal of positions but also the orientations inside the space.

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.

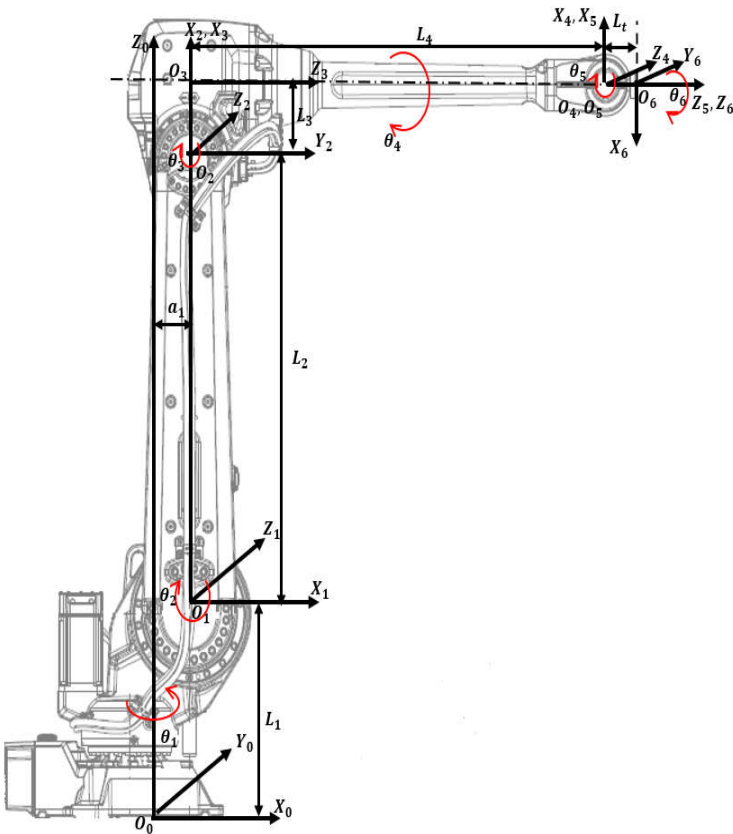


Figure A1. IRB ABB 4600 Model with attached frames.

Dimensions are in mm

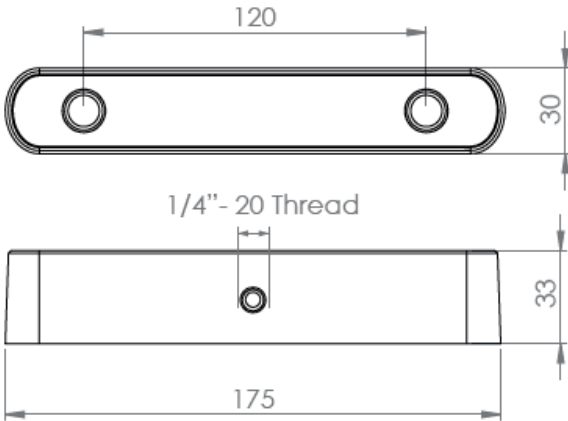


Figure A2. Zed 2 stereo camera model with dimensions.

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: $W \times H$	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.

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:

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

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

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

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

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

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:

$$\begin{aligned} p_x &= d_x - L_t a_x \\ p_y &= d_y - L_t a_y \end{aligned} \quad (B6)$$

$$\begin{aligned} p_z &= d_z - L_t a_z \\ q_1 &= \arctan\left(\frac{p_y}{p_x}\right) \end{aligned} \quad (B7)$$

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

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

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

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

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

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

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. W. Chang and C.-H. Wu, "Automated USB peg-in-hole assembly employing visual servoing," in Proc. 3rd Int. Conf. Control, Autom. Robot. (ICCAR). Nagoya, Japan: IEEE. Apr. 2017, pp. 352–355.
2. J. Xu, K. Liu, Y. Pei, C. Yang, Y. Cheng, and Z. Liu, "A noncontact control strategy for circular peg-in-hole assembly guided by the 6-DOF robot based on hybrid vision," IEEE Trans. Instrum. Meas., vol. 71, 2022, Art. no. 3509815.
3. W. Zhu, H. Liu, and Y. Ke, "Sensor-based control using an image point and distance features for rivet-in-hole insertion," IEEE Trans. Ind. Electron., vol. 67, no. 6, pp. 4692–4699, Jun. 2020.
4. F. Qin, D. Xu, D. Zhang, W. Pei, X. Han and S. Yu, "Automated Hooking of Biomedical Microelectrode Guided by Intelligent Microscopic Vision," in IEEE/ASME Transactions on Mechatronics, vol. 28, no. 5, pp. 2786–2798, Oct. 2023, doi: 10.1109/TMECH.2023.3248112.
5. Jung, Woo, Jin., Lee, Hyun, Jung. (2020). Automatic alignment type welding apparatus and welding method using the above auto-type welding apparatus.
6. P. Guo, Z. Zhang, L. Shi, and Y. Liu, "A contour-guided pose alignment method based on Gaussian mixture model for precision assembly," Assem. Autom., vol. 41, no. 3, pp. 401–411, Jul. 2021.
7. F. Chaumette and S. Hutchinson, "Visual servo control Part I: Basic approaches," IEEE Robotics & Automation Magazine, vol. 13, pp. 82–90, 2006.
8. 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.
9. 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.
10. G. Flandin, F. Chaumette and E. Marchand, "Eye-in-hand/eye-to-hand cooperation for visual servoing," Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA, 2000, pp. 2741–2746 vol.3, doi: 10.1109/ROBOT.2000.846442.
11. Muis, A., & Ohnishi, K. (n.d.). Eye-to-hand approach on eye-in-hand configuration within real-time visual servoing. The 8th IEEE International Workshop on Advanced Motion Control, 2004. AMC '04. doi:10.1109/amc.2004.1297945

12. A. Ruf, M. Tonko, R. Horaud, and H.-H. Nagel: "Visual Tracking of An End-Effector by Adaptive Kinematic Prediction", Proc. of the Intl. ConJ on Intelligent Robots and Systems(IROS'97). vol. 2, pp.893-898,Grenoble, France, Sep.(1997).
13. Jose Luis de Mena: "Virtual Environment for Development of Visual Servoing Control Algorithms", Thesis, Lund Institute Technology, Sweden, May, (2002).
14. K. Irie, I. M. Woodhead, A. E. McKinnon and K. Unsworth, "Measured effects of temperature on illumination-independent camera noise," 2009 24th International Conference Image and Vision Computing New Zealand, Wellington, New Zealand, 2009, pp. 249-253, doi: 10.1109/IVCNZ.2009.5378403.
15. 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.
16. Das, S., Saikia, J., Das, S., & Goñi, N. A COMPARATIVE STUDY OF DIFFERENT NOISE FILTERING TECHNIQUES IN DIGITAL IMAGES. 2015.
17. 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.
18. 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
19. G. Chen, F. Zhu and P. A. Heng. An Efficient Statistical Method for Image Noise Level Estimation. In: 2015 IEEE International Conference on Computer Vision (ICCV); 2015.p. 477-485. DOI: 10.1109/ICCV.2015.62.
20. Mark, W.S., M.V. (1989). Robot Dynamics and control. John Wiley & Sons, Inc. 1989.
21. W. W. MARZOUK, A. ELBASET etc. (2019)." Modelling, Analysis and Simulation for A 6 Axis Arm Robot by PID Controller", International Journal of Mechanical and Production Engineering Research and Development (IJMPERD) ISSN (P): 2249-6890; ISSN (E): 2249-8001 Vol. 9, Issue 4, Aug 2019, pp. 363-376. DOI: <https://doi.org/10.24247/ijmperd201937>.
22. Li, R., & Assadian, F. Role of Uncertainty in Model Development and Control Design for a Manufacturing Process. In: Majid T., Pengzhong L, & Liang L, editors. Production Engineering and Robust Control. London: IntechOpen 2022. pp. 137-167. DOI:10.5772/intechopen.101291
23. Anonymous. ABB IRB 4600 -40/2.55 Product Manual [Internet]. 2013. Available from: <https://www.manualslib.com/manual/1449302/Abb-Irb-4600-40-2-55.html#manual> [Accessed: 2024-7-17]
24. Anonymous. Stereolabs Docs: API Reference, Tutorials, and Integration. Available from: <https://www.stereolabs.com/docs> [Accessed: 2024-7-18]
25. Peer, P., & Solina, F. (2006). Where physically is the optical center? Pattern Recognition Letters, 27(10), 1117–1121. doi:10.1016/j.patrec.2005.12.
26. Vassilios D Tourassis, Dimitrios M Emiris. (1993) "A comparative study of ideal elbow and dual-elbow robot manipulators", Mechanism and Machine Theory, Volume 28, Issue 3, Pages 357-373, ISSN 0094-114X, [https://doi.org/10.1016/0094-114X\(93\)90076-8](https://doi.org/10.1016/0094-114X(93)90076-8).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.