2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

LLM-Guided Semantic Stereo Adaptive Visual Servoing for Precise Peg-in-Hole

Xiyue Dong, Guangli Sun, Jinfei Hu, Tianyu Huang, Wei Chen, Yun-Hui Liu

***Abstract*— Precision assembly tasks like peg-in-hole remain challenging for robotic manipulation. While visual servoing offers a robust framework, it depends heavily on accurate calibration and manual feature engineering. Learning-based methods, including vision-language models (VLMs), provide strong semantic understanding but often lack the precision needed for high-tolerance, contact-rich insertions. This paper introduces a novel framework that combines the semantic reasoning of large language models (LLMs) with adaptive visual servoing to bridge this gap. Our approach uses an LLM as a semantic feature extractor and correspondence engine for stereo visual servoing. The LLM processes generic point features from uncalibrated stereo images along with a task description in natural language, leveraging its spatial understanding to identify and correspond optimal features across views. These features drive a stereo adaptive visual servoing controller that estimates unknown calibration parameters online, enabling precise, calibration-free positioning. Extensive evaluations on cylindrical, square, and hexagonal peg-in-hole tasks across three trials demonstrate average success rates above 90% with steady-state errors of 1.8–2.8 pixels, closely comparable to calibrated methods (1.2–2.5 pixels). This is achieved without requiring prior models, calibration, or task-specific training, thereby advancing flexible and precise robotic assembly.**

I. INTRODUCTION

Robotic peg-in-hole insertion is a fundamental yet challenging task in industrial automation and dexterous manipulation [1]. While force-based strategies have been widely adopted due to their robustness to positional uncertainties [2], they often result in prolonged contact-rich interactions that reduce efficiency. Visual and hybrid visual-force servoing methods offer a more direct and potentially faster alternative by leveraging exteroceptive sensing to guide the peg prior to and during insertion [3]. However, these approaches typically require highly accurate camera calibration, precise object models, and manual feature engineering—such as defining and extracting specific geometric features in advance [4], which severely limits their flexibility and practicality in unstructured settings. Enabling efficient and high-precision peg-in-hole tasks without prior models or calibrated cameras remains an open research problem [5]. Recent advances in unseen object pose estimation have explored learning-based deformation of geometric primitives to achieve model-free pose prediction for novel objects [6], yet integrating such representations directly into real-time closed-loop servoing for contact-rich tasks remains challenging.

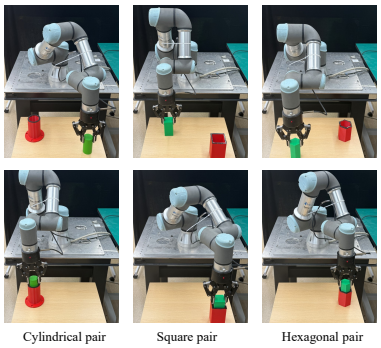

Fig. 1: Our framework leverages large language models as semantic feature extractors for adaptive stereo visual servoing, enabling accurate, calibration-free manipulation in precise assembly tasks.

Classical visual servoing techniques depend on accurately identified visual features (e.g., points, edges, or corners) and a well-calibrated camera system to achieve convergence [7]. These methods are mathematically sound and perform robustly in controlled environments. However, they are sensitive to errors in camera intrinsic and extrinsic parameters, and require meticulous hand-designing of features for each new object or task [8]. This process often involves significant human intervention, including manual labeling and trial-and-error tuning, which becomes impractical when dealing with diverse or unfamiliar assembly setups [9].

With the emergence of large-scale vision-language models (VLMs) and vision-based policies, there has been growing interest in leveraging pre-trained models for semantic scene understanding and end-to-end control [10]. These methods exhibit remarkable generalizability and can interpret open-ended instructions, allowing robots to operate in more varied environments without task-specific programming [11]. However, while they excel in semantic reasoning and coarse manipulation, they often fall short in delivering the high-precision control required for tight-tolerance tasks such as peg-in-hole [12]. Their performance is constrained by training data quality and the inherent stochasticity of generative models, making them

This work is supported in part by the Research Grants Council (RGC) of Hong Kong under Grant 14218322, in part by the HK RGC AoE under AoE/E-407/24-N, in part by the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government via the Hong Kong Centre for Logistics Robotics, and in part by the CUHK T Stone Robotics Institute. (Corresponding author: Yun-Hui Liu).

The authors are all with the T Stone Robotics Institute, Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, SAR, and the Hong Kong Centre for Logistics Robotics, Hong Kong, SAR. (e-mail: 1155166244@link.cuhk.edu.hk; guanglisun@cuhk.edu.hk; jinfeihu@cuhk.edu.hk; tianyuhuang@cuhk.edu.hk; weichen@link.cuhk.edu.hk; yhliu@mae.cuhk.edu.hk).

unreliable for millimeter-level accuracy [13].

To overcome these limitations, we propose a novel framework that integrates the semantic reasoning capability of large language models (LLMs) with stereo-based adaptive visual servoing. Our method first extracts generic point features from both the peg and hole objects across stereo images without relying on predefined geometric models or calibrated cameras. These visual features, along with a natural language task description, are provided to an LLM, which leverages its inherent understanding of spatial relationships and assembly contexts to infer and correspond the most relevant feature points between the two views for servoing. The selected features are then used in a stereo-based adaptive visual servoing controller, which simultaneously estimates the unknown camera parameters and achieves accurate 3D positioning through a unified adaptive law across both views. This combination allows the system to perform high-precision insertions without prior knowledge of object geometry or camera parameters, explicitly addressing the calibration challenge in stereo vision.

The main contributions of this work are as follows:

1) We introduce a new LLM-guided visual servoing framework that translates semantic task descriptions into actionable feature correspondences for precision manipulation.
2) We develop a calibration-free stereo visual servoing system that uses adaptive control to achieve robust peg-in-hole insertion with a tolerance under 3mm.
3) We validate our approach extensively on both cylindrical and square peg-in-hole tasks, demonstrating superior performance without prior models or camera calibration.

RELATED WORK

A. Learning-Based Visual Manipulation

The rise of deep learning has spurred the development of end-to-end visuomotor policies that learn to map raw pixels directly to robot actions [13]. Imitation learning (IL) [14] and reinforcement learning (RL) [15] have achieved remarkable success in tasks like grasping and dexterous manipulation [16]. More recently, *Diffusion Policies* [17] have emerged as a powerful generative approach for producing multimodal, robust behavior. While these methods reduce the need for explicit feature design and show strong generalization across object appearances and backgrounds, they are notoriously data-hungry [18] and often struggle to achieve the sub-millimeter precision required for high-tolerance assembly tasks [19]. Their "black-box" nature also makes it difficult to diagnose failures or incorporate safety guarantees. Our method differs by not learning a full end-to-end policy. Instead, we use a pre-trained LLM as a semantic feature extractor, preserving the interpretability and precision of classical control loops while gaining the generalization benefits of a large-scale model.

B. LLM-Guided Robot Manipulation

Large Language Models (LLMs) and Vision-Language Models (VLMs) [20] have revolutionized high-level robot planning and semantic reasoning. Foundational works like *SayCan* [21] demonstrated that LLMs can break down complex natural language instructions into structured action sequences or executable code by leveraging their embedded commonsense knowledge. Subsequent research has focused on grounding these plans in physical reality, using VLMs to identify objects and their affordances from images [22]. For instance, *VoxPoser* [22] uses an LLM to compose 3D value maps for reward shaping, guiding policies towards task completion. However, these approaches primarily operate at the symbolic level of *what* to do, delegating the low-level *how* to execute the motion to a separate controller, which is often still traditional. The key insight of our work is to push LLM guidance deeper into the control stack, using it not just for task planning but to directly inform the perceptual input (feature selection) for a low-level visual servo controller.

C. Adaptive Visual Servoing

Conventional Visual servoing (VS) requires laborious extrinsic calibration, which must be repeated whenever the camera setup changes [7]. To overcome such calibration challenges, adaptive visual servoing has been extensively studied, drawing inspiration from classical adaptive control methods developed for robotic systems [23]–[25]. Early works introduced online estimation of the image Jacobian [26] or depth-independent interaction matrices [27], enabling control with uncalibrated or time-varying camera parameters. Subsequent studies generalized the framework of depth-independent interaction matrix to contour-based features [28], shape control [29] and constraint-aware designs [30]. Most approaches, typically grounded in the Slotine-Li adaptive control framework [31], operate either at the kinematic level (velocity control) [29] or the dynamic level (torque control) [27]. While these methods significantly improve robustness, they continue to rely on pre-defined, hand-engineered features such as corners or edges.

Recently, a new direction has emerged by incorporating semantics into VS. Instead of low-level geometric features, semantic segmentation masks or object detectors have been explored as higher-level visual cues [32]. With the advent of foundation models, this idea has evolved toward zero-shot visual servoing, where vision-language models (VLMs) generate goal images, textual descriptions, or spatial keypoints for control [33]. Despite their potential, these approaches often demand extensive feature-related training. Our work builds on this line of research by introducing *semantic-adaptive visual servoing*, which leverages an LLM's reasoning ability to interpret task context (e.g., peg-in-hole) and dynamically select relevant point features from generic visual input. This eliminates the need for manual feature engineering, object models, or task-specific training, while retaining the robustness of adaptive control.

II. METHODOLODGY

A. Problem Statement

This work investigates the visual servoing problem for high-precision manipulation under significant parametric uncertainty. Consider a robotic system performing peg-in-hole assembly tasks guided by an uncalibrated stereo vision system.

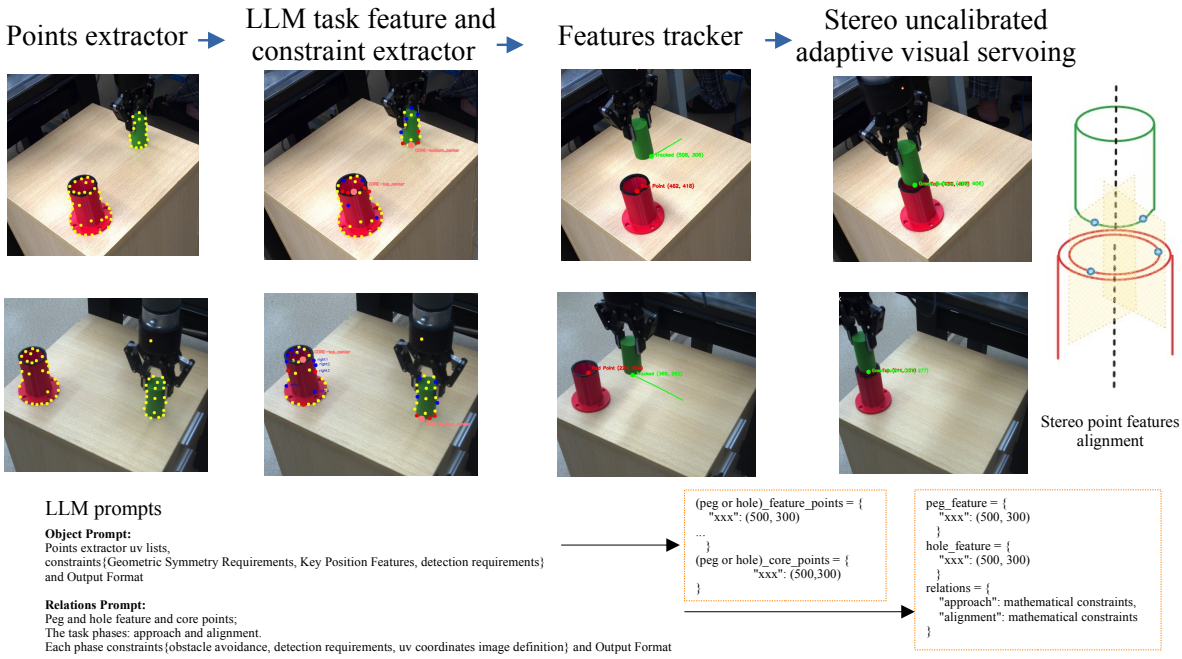

Fig. 2: LLM-guided semantic stereo adaptive visual servoing framework. The framework utilizes LLM-generated object prompts to identify task-specific feature points from raw UV coordinates extracted by a point extractor in the first stereo camera frames. These feature points are then tracked in real-time during stereo adaptive visual servoing. Both the peg-in-hole approach and alignment phases are guided by constraints derived from LLM relation prompts, ensuring collision avoidance and reliable feature detection. The relation-aware stereo alignment enables collision-free insertion based on stereo-axis correspondence, with gripper release completing the assembly.

The fundamental challenge lies in the unknown projective geometry: neither the intrinsic parameter matrix $\bar{\boldsymbol{\Omega}} \in \mathbb{R}^{3\times3}$ nor the extrinsic transformation ${}^{c}\boldsymbol{T}_{b} \in SE(3)$ between camera frame Σ_c and robot base frame Σ_b is known.

The visual servoing kinematics establish a nonlinear mapping between the robot's configuration space and image feature space:

$$\boldsymbol{X}_p = \pi(\bar{\boldsymbol{\Omega}}, {}^{c}\boldsymbol{T}_{b}, {}^{b}\boldsymbol{T}_{e}, \boldsymbol{X}_e) \tag{1}$$

where $\pi : \mathbb{R}^{3\times3} \times SE(3) \times SE(3) \times \mathbb{R}^3 \to \mathbb{R}^2$ represents the projective transformation, and $\boldsymbol{X}_p = [u_p, v_p]^T \in \mathbb{R}^2$, $\boldsymbol{X}_e \in \mathbb{R}^3$ denote feature positions in image and end-effector coordinates, respectively.

The critical challenge in peg-in-hole assembly lies in achieving precise axial alignment between peg and hole. For successful insertion with clearances under 3mm, the desired image feature coordinates $\boldsymbol{X}_p^*$ must be determined with sub-pixel accuracy. However, manually defining these precise feature points is infeasible due to calibration uncertainties and variations in object appearance.

To address this fundamental limitation, we propose a novel approach that leverages stereo vision and large language models. The precise feature coordinates $\boldsymbol{X}_p^*$ are mapped to the stereo camera system, where the LLM semantically identifies optimal alignment features across both views. Formally, the LLM learns a mapping:

$$\Phi : (\boldsymbol{I}_L, \boldsymbol{I}_R, \mathcal{T}) \to (\boldsymbol{X}_{p,L}^*, \boldsymbol{X}_{p,R}^*) \tag{2}$$

where $\boldsymbol{I}_L, \boldsymbol{I}_R$ represent left and right stereo images, $\mathcal{T}$ is a natural language task description, and $\boldsymbol{X}_{p,L}^*, \boldsymbol{X}_{p,R}^*$ are the semantically-derived target feature coordinates.

When the visual servoing controller achieves alignment such that:

$$\|\boldsymbol{X}_{p,L} - \boldsymbol{X}_{p,L}^*\| < \varepsilon \quad \text{and} \quad \|\boldsymbol{X}_{p,R} - \boldsymbol{X}_{p,R}^*\| < \varepsilon \tag{3}$$

for some small threshold ε, the peg-hole system attains the required axial alignment for successful insertion. This approach transforms the precision alignment problem from explicit geometric calibration to semantic feature understanding, enabling robust peg-in-hole assembly without prior calibration or manual feature engineering.

The LLM-guided semantic stereo adaptive visual servoing framework is proposed to solve the peg-in-hole problem without any calibration parameters and model information as shown in Fig. 2. The framework utilizes LLM-generated object prompts to identify task-specific feature points from raw UV coordinates extracted by a point extractor in the first stereo camera frames. These feature points are then tracked in real-time during stereo adaptive visual servoing. Both the peg-in-hole approach and alignment phases are guided by constraints

derived from LLM relation prompts, ensuring collision avoidance and reliable feature detection. The sample LLM prompt can be seen in Appendix.

B. LLM-Guided Feature and Relation Extraction

To address the challenges of precise feature identification and constraint derivation under significant projective uncertainty, we introduce an LLM-based semantic feature and relation extraction module. This module leverages the contextual and geometric reasoning capabilities of large language models to generate task-specific feature points and relational constraints from raw stereo images, without relying on explicit calibration or manual engineering.

1) Semantic Feature Point Extraction: Let $\boldsymbol{I}_L, \boldsymbol{I}_R \in \mathbb{R}^{H\times W\times 3}$ denote the left and right images from the stereo vision system. A classical feature point extractor (e.g., SIFT or ORB) is first applied to both images to obtain a set of candidate points:

$$\mathcal{P}_L = \{(u_i^L, v_i^L)\}_{i=1}^{N_L}, \quad \mathcal{P}_R = \{(u_j^R, v_j^R)\}_{j=1}^{N_R}. \tag{4}$$

These raw points are often noisy, redundant, and lack semantic relevance to the task. To filter and select task-relevant features, we formulate a prompt $\mathcal{T}_{\text{object}}$ that encodes geometric and semantic constraints specific to peg-in-hole assembly. The prompt is designed to guide the LLM in selecting points that satisfy:

- Oblique inner edge distribution,
- Proximal visibility under slant,
- Geometric symmetry along the inclined axis,
- Stability and representativeness throughout insertion.

The LLM processes the prompt along with the raw point sets and outputs a refined set of feature points:

$$\Phi_{\text{object}}\colon (\mathcal{P}_L, \mathcal{P}_R, \mathcal{T}_{\text{object}}) \to (\mathcal{F}_L, \mathcal{F}_R), \tag{5}$$

where $\mathcal{F}_L = \{(u_k^L, v_k^L)\}_{k=1}^{K}$, $\mathcal{F}_R = \{(u_m^R, v_m^R)\}_{m=1}^{M}$ with $K, M \le 8$, and a subset of core points $\mathcal{C}_L, \mathcal{C}_R$ (each with 4 points) for robust tracking.

The selection process can be expressed as an optimization problem:

$$\max_{\mathcal{F}\subseteq\mathcal{P}} \sum_{p\in\mathcal{F}} w_{\text{sym}}(p) + w_{\text{stab}}(p) + w_{\text{prox}}(p), \tag{6}$$

subject to geometric constraints such as symmetry along the oblique axis and proximity to the inner edge.

2) Relation Prompt for Constraint Derivation: During visual servoing, relational constraints between peg and hole features are critical to avoid collisions and ensure alignment. We define a relation prompt $\mathcal{T}_{\text{relation}}$ that instructs the LLM to output a pair of feature points (one from peg, one from hole) and their constraints for each phase: approach and alignment.

Let

$$\boldsymbol{p}_{\text{peg}} = (u_p, v_p), \quad \boldsymbol{p}_{\text{hole}} = (u_h, v_h). \tag{7}$$

The LLM outputs a structured relation tuple:

$$\Phi_{\text{relation}}\colon (\mathcal{F}_L, \mathcal{F}_R, \mathcal{T}_{\text{relation}}) \to \big(\boldsymbol{p}_{\text{peg}}, \boldsymbol{p}_{\text{hole}}, \mathcal{R}_{\text{approach}}, \mathcal{R}_{\text{alignment}}\big), \tag{8}$$

where $\mathcal{R}_{\text{approach}}$ and $\mathcal{R}_{\text{alignment}}$ are mathematical constraints expressed in image coordinates. For example:

- **Approach Phase**: Avoid collision by maintaining

$$v_p < v_h - \delta_v, \tag{9}$$

  ensuring the peg is above the hole in the image plane.
- **Alignment Phase**: Achieve sub-pixel alignment along the inclined axis:

$$|u_p - u_h| < \varepsilon_u, \quad |v_p - v_h| < \varepsilon_v. \tag{10}$$

These constraints are derived from the LLM's understanding of spatial relationships under projective geometry, leveraging natural-language descriptions of relative positions (e.g., "above", "left", "aligned").

3) Integration with Visual Servoing: The selected feature points and constraints are passed to the stereo adaptive visual servoing controller. The real-time feature tracker (e.g., KLT) maintains correspondence of $\mathcal{F}_L$ and $\mathcal{F}_R$ across frames. The controller then minimizes the error

$$\boldsymbol{e}(t) = \begin{bmatrix} \boldsymbol{X}_{p,L}(t) - \boldsymbol{X}_{p,L}^* \\ \boldsymbol{X}_{p,R}(t) - \boldsymbol{X}_{p,R}^* \end{bmatrix}, \tag{11}$$

where $\boldsymbol{X}_{p,L}^*$, $\boldsymbol{X}_{p,R}^*$ are the LLM-generated target features. The relation constraints $\mathcal{R}$ are incorporated as inequality or equality constraints in the control law to ensure collision-free and aligned motion.

This LLM-guided approach transforms the problem from explicit geometric calibration to semantic understanding, enabling robust and precise peg-in-hole assembly under significant uncertainty.

C. Stereo Uncalibrated Adaptive Visual Servoing

1) Kinematic Model: The detailed expression of (1) in the pinhole model with a fixed camera is as follows. A 3-D point fixed to the robot end-effector frame ${}^e\boldsymbol{x} = [\boldsymbol{X}_e^T, 1]^T \in \mathbb{R}^4$ is projected to image pixels $\boldsymbol{y} = [\boldsymbol{X}_p^T, 1]^T \in \mathbb{R}^3$ by

$$\boldsymbol{y} = \frac{1}{{}^cz(\boldsymbol{q}(t))}\,\boldsymbol{\Omega}\,{}^c\boldsymbol{T}_b\,{}^b\boldsymbol{T}_e(\boldsymbol{q})\,{}^e\boldsymbol{x}, \tag{12}$$

where

- $\boldsymbol{\Omega} = [\bar{\boldsymbol{\Omega}}, \boldsymbol{0}_{3\times 1}] \in \mathbb{R}^{3\times 4}$ is the augmented intrinsic matrix,
- ${}^c\boldsymbol{T}_b \in \mathbb{R}^{4\times 4}$ is the extrinsic matrix,
- ${}^b\boldsymbol{T}_e(\boldsymbol{q}) \in \mathbb{R}^{4\times 4}$ is the forward kinematics from the end-effector frame Σ_e to the robot base frame, parameterized by joint variables $\boldsymbol{q}(t)$,
- ${}^cz(\boldsymbol{q}(t))$ denotes the depth of the point in the camera frame.

Let $\boldsymbol{M} = \boldsymbol{\Omega}\,{}^c\boldsymbol{T}_b$ be a constant matrix, and denote its i-th row by $\boldsymbol{m}_i^T$, then the depth and its time derivative are

$$\begin{aligned} {}^cz(\boldsymbol{q}(t)) &= \boldsymbol{m}_3^T\,{}^b\boldsymbol{T}_e(\boldsymbol{q})\,{}^e\boldsymbol{x}, \\ {}^c\dot{z}(\boldsymbol{q}(t)) &= \boldsymbol{m}_3^T\,{}^b\dot{\boldsymbol{T}}_e(\boldsymbol{q})\,{}^e\boldsymbol{x} = \boldsymbol{m}_3^T \begin{bmatrix} skew(\boldsymbol{\omega})\,{}^b\boldsymbol{R}_e\boldsymbol{X}_e + \boldsymbol{v} \\ 0 \end{bmatrix} \end{aligned} \tag{13}$$

where $\boldsymbol{v}$ and $\boldsymbol{\omega}$ represent the linear and angular velocity of the robot end-effector, and *skew* denotes the skew-symmetric operator.

The differential kinematics is obtained according to (12) as

$$\dot{\boldsymbol{y}} = -\frac{{}^c\dot{z}}{{}^cz^2}\boldsymbol{M}^b\boldsymbol{T}_e{}^e\boldsymbol{x} + \frac{1}{{}^cz}\boldsymbol{M}^b\dot{\boldsymbol{T}}_e{}^e\boldsymbol{x} = \frac{1}{{}^cz}\boldsymbol{A}(\boldsymbol{y})\boldsymbol{J}_x\begin{bmatrix}\boldsymbol{v}\\ \boldsymbol{\omega}\end{bmatrix} \tag{14}$$

with

$$\begin{aligned}\boldsymbol{A}(\boldsymbol{y}) &= \boldsymbol{M} - \boldsymbol{y}\boldsymbol{m}_3^T = \begin{bmatrix}\boldsymbol{m}_1^T - u_p\boldsymbol{m}_3^T\\ \boldsymbol{m}_2^T - v_p\boldsymbol{m}_3^T\\ \boldsymbol{0}_{1\times 4}\end{bmatrix}\\ \boldsymbol{J}_x &= \begin{bmatrix}\boldsymbol{I}_{3\times 3} & -skew({}^b\boldsymbol{R}_e\boldsymbol{X}_e)\\ \boldsymbol{0}_{1\times 3} & \boldsymbol{0}_{1\times 3}\end{bmatrix}\\ \begin{bmatrix}\boldsymbol{v}\\ \boldsymbol{\omega}\end{bmatrix} &= \boldsymbol{J}\dot{\boldsymbol{q}}\end{aligned} \tag{15}$$

where $\boldsymbol{A}(\boldsymbol{y})$ is called the depth-independent interaction matrix; $\boldsymbol{J}$ is the robot Jacobian matrix.

2) Controller Design: The controller for a single camera setup is designed as

$$\dot{\boldsymbol{q}} = -(\boldsymbol{J}_x\boldsymbol{J})^\dagger(\hat{\boldsymbol{A}}^T(\boldsymbol{y}) + \tfrac{1}{2}\hat{\boldsymbol{m}}_3(t)\Delta\boldsymbol{y}^T(t))\boldsymbol{K}\Delta\boldsymbol{y}(t) \tag{16}$$

where $\Delta\boldsymbol{y} = \boldsymbol{y} - \boldsymbol{y}^*$ denotes the control error with $\boldsymbol{y}^*$ the desired feature position; $\boldsymbol{K} \succ 0$ is the control gain matrix. The matrix $\boldsymbol{M}$ contains 12 elements, of which only 11 are independent [27]. As the intrinsic and extrinsic parameters are not calibrated, $\boldsymbol{M}$ is uncertain and parameterized using the stacked vector $\boldsymbol{\theta} = \text{vec}(\boldsymbol{M})$. Its estimates and estimation error are denoted by $\hat{\boldsymbol{\theta}} = \text{vec}(\hat{\boldsymbol{M}})$ and $\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$ respectively. Based on $\hat{\boldsymbol{\theta}}(t)$, the estimated projection matrix is $\hat{\boldsymbol{M}}(t)$ and the estimated interaction matrix is given by $\hat{\boldsymbol{A}}(\boldsymbol{y}) = \hat{\boldsymbol{M}} - \hat{\boldsymbol{y}}\boldsymbol{m}_3^T$.

Defining $\boldsymbol{D}(t) = \boldsymbol{A}^T(\boldsymbol{y}) + \frac{1}{2}\boldsymbol{m}_3^T\Delta\boldsymbol{y}$, a linear regression with respect to the parameter estimation error can be obtained in the following form,

$$\Delta\boldsymbol{y}^T\boldsymbol{K}\tilde{\boldsymbol{D}}^T\hat{\boldsymbol{D}}\boldsymbol{K}\Delta\boldsymbol{y} = \boldsymbol{Y}(\boldsymbol{y}(t), \hat{\boldsymbol{\theta}}(t))\Delta\boldsymbol{\theta}(t), \tag{17}$$

Based on the linear regression (17), the adaptation law is designed as

$$\dot{\hat{\boldsymbol{\theta}}}(t) = -\boldsymbol{\Gamma}^{-1}\boldsymbol{Y}^T(\boldsymbol{y}(t), \hat{\boldsymbol{\theta}}(t)) \tag{18}$$

where $\boldsymbol{\Gamma} \succ 0$ is the adaptation rate matrix.

To prove the stability of the closed-loop system, consider the following Lyapunov candidate

$$V = \tfrac{1}{2}{}^cz(\boldsymbol{q}(t))\Delta\boldsymbol{y}(t)^T\boldsymbol{K}\Delta\boldsymbol{y}(t) + \tfrac{1}{2}\Delta\boldsymbol{\theta}(t)^T\boldsymbol{\Gamma}\Delta\boldsymbol{\theta}(t), \tag{19}$$

Substituting the control law (16) and the adaptation law (18), and noting (13), the time derivative of V is

$$\begin{aligned}\dot{V} &= {}^cz(\boldsymbol{q}(t))\Delta\boldsymbol{y}^T\boldsymbol{K}\dot{\boldsymbol{y}} + \tfrac{1}{2}{}^c\dot{z}(\boldsymbol{q}(t))\Delta\boldsymbol{y}^T\boldsymbol{K}\Delta\boldsymbol{y} + \Delta\boldsymbol{\theta}^T\boldsymbol{\Gamma}\Delta\dot{\boldsymbol{\theta}}\\ &= -\Delta\boldsymbol{y}^T\boldsymbol{K}\hat{\boldsymbol{D}}^T\hat{\boldsymbol{D}}\boldsymbol{K}\Delta\boldsymbol{y} - \Delta\boldsymbol{y}^T\boldsymbol{K}\hat{\boldsymbol{D}}^T\hat{\boldsymbol{D}}\dot{\boldsymbol{x}} + \Delta\boldsymbol{\theta}^T\boldsymbol{\Gamma}\Delta\dot{\boldsymbol{\theta}}\\ &= \Delta\boldsymbol{y}^T\boldsymbol{K}\hat{\boldsymbol{D}}^T\hat{\boldsymbol{D}}\boldsymbol{K}\Delta\boldsymbol{y} \le 0\end{aligned} \tag{20}$$

where $\dot{\boldsymbol{x}} = {}^b\dot{\boldsymbol{T}}_e{}^e\boldsymbol{x}$. By Babrrat's Lemma, it could be concluded that $\lim_{t\to\infty}\hat{\boldsymbol{D}}\boldsymbol{K}\Delta\boldsymbol{y} = \boldsymbol{0}$. As $\hat{\boldsymbol{D}}$ is of rank 2 [27], $\Delta\boldsymbol{y} \to 0$ as $t \to \infty$.

For the dual-camera setup, once the feature point converges to its desired position in each image frame, it also converges to the desired position in 3D Cartesian space. Accordingly, in adaptive stereo visual servoing with two cameras, the control law is designed as

$$\begin{aligned}\dot{\boldsymbol{q}} = -(\boldsymbol{J}_x\boldsymbol{J})^\dagger\Big[&(\hat{\boldsymbol{A}}_l{}^T + \tfrac{1}{2}\hat{\boldsymbol{m}}_{3,l}\Delta\boldsymbol{y}_l^T)\boldsymbol{K}\Delta\boldsymbol{y}_l\\ &+(\hat{\boldsymbol{A}}_r{}^T + \tfrac{1}{2}\hat{\boldsymbol{m}}_{3,r}\Delta\boldsymbol{y}_r^T)\boldsymbol{K}\Delta\boldsymbol{y}_r\Big]\end{aligned} \tag{21}$$

with the adaptation law

$$\begin{aligned}\dot{\hat{\boldsymbol{\theta}}}_l(t) &= -\boldsymbol{\Gamma}^{-1}\boldsymbol{Y}_l(\boldsymbol{y}_l, \hat{\boldsymbol{\theta}}_l)^T\\ \dot{\hat{\boldsymbol{\theta}}}_r(t) &= -\boldsymbol{\Gamma}^{-1}\boldsymbol{Y}_r(\boldsymbol{y}_r, \hat{\boldsymbol{\theta}}_r)^T\end{aligned} \tag{22}$$

where the subscript l and r correspond to the left and right cameras, respectively.

Fig. 3: Experimental setup with left and right uncalibrated FLIR pinhole RGB cameras

III. Experiments

A. Experimental Setup

To validate the proposed adaptive dual uncalibrated hand-to-eye visual servoing (VS) framework with LLM-guided geometric feature alignment, we employ a simple peg-in-hole task using a 6-DOF Universal Robots UR3e robotic arm equipped with a Robotiq 2F-85 gripper. Two uncalibrated FLIR BFS-U3-32S4C-C RGB cameras mounted on tripods with a 0.4 m baseline to ensure sufficient disparity and full workspace coverage. To demonstrate generality across geometries, we test three peg-hole pairs: a green cylindrical peg (diameter 40 mm) with a red cylindrical hole (inner diameter 45 mm); a green square peg (40 mm x 40 mm) with a red square hole (45 mm x 45 mm); and a green hexagonal peg (40 mm side length) with a red hexagonal hole (45 mm side length). For each pair, three trials evaluate performance across different configurations (see Fig. 3). Each trial includes 10 repetitions, measuring success by collision-free successful insertion and final pixel-converged steady state error (SSE). Feature identification and tracking follow a general and calibration-free pipeline: HSV thresholding segments pegs (green hue: 60–120 degrees) and holes (red hue: 0–10 degrees or 170–180 degrees) for robust isolation; The AKAZE corner detection algorithm, selected for its high repeatability and distinctiveness of detected features across stereo views, extracts initial corner lists from the starting stereo frames; Deepseek R2 (LLM module) processes these multiple corner coordinates along with a general task-semantic description to establish constraints and refine to a single, task-relevant 2D pixel point pair; Lucas-Kanade (LK) optical flow,

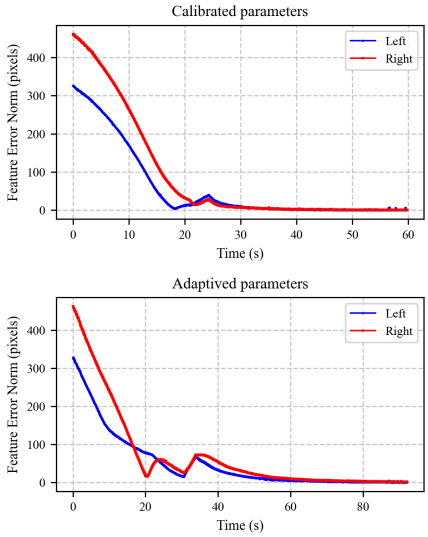

Fig. 4: Feature error over time using traditional and adaptive controllers for cylindrical peg-hole pairs in Trial 1, which shows the adaptive controller has the same convergence efficiency as the calibrated camera parameters.

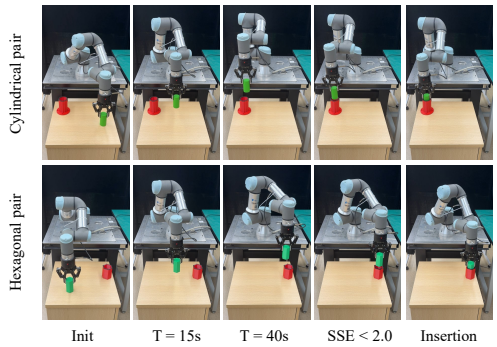

Fig. 5: Snapshots for cylindrical and hexagonal pairs' experiments. It achieves 2-pixel peg-hole alignment precision

selected for its accuracy and efficiency in handling small inter-frame motions, tracks them at 30 Hz in real control. Image-space errors between peg and hole features drive the adaptive IBVS controller to compute Cartesian end-effector velocities for alignment. To maintain vertical alignment between the peg and hole during feature point alignment, the controller outputs only linear end-effector velocities for large image-space errors' norm (>100 pixels), incorporating angular velocities for fine-tuning only when errors are small.

Images are streamed at 30 Hz via ROS2. After extracting the initial coordinates of the peg-hole feature points from the LLM, these points are continuously tracked, and the 2D image-space errors between the peg and hole features are computed and fed to the controller. The robot arm is driven by joint velocities via the control law (16). We also conducted experiments using a traditional IBVS controller for comparison, which requires prior calibration of camera intrinsic and extrinsic parameters via standard checkerboard methods.

B. Experimental results

To evaluate the performance of the proposed adaptive dual uncalibrated hand-to-eye VS framework, we analyze convergence behavior and task success across the three peg-hole pairs (cylindrical, square, and hexagonal) in the simplified vertical-alignment setup. All trials demonstrate robust feature alignment and insertion, validating our proposed framework's efficacy for semantic-guided geometric tasks.

Fig. 4 illustrates the image-space error convergence for representative cylindrical peg-hole trials, comparing the proposed adaptive IBVS controller against the traditional calibrated baseline. The traditional controller, relying on pre-calibrated intrinsics and extrinsics, converges to a steady-state SSE of 1–3 pixels. In contrast, the adaptive controller achieves convergence to 3–5 pixels. This minor performance gap underscores the adaptive framework's ability to attain sub-pixel precision without calibration overhead.

Task success rates exceed 90 percent across 90 total repetitions (10 per trial, 3 trials per pair), with the majority of failures due to LK tracking errors and a minimal portion attributable to initial corner selections that preclude successful insertion. Fig. 5 illustrates the temporal evolution of a representative peg-in-hole and successful insertions across diverse peg-hole pairs. These visuals affirm the framework's practicality: the LLM's semantic selection of task-relevant points (e.g., centroids guided by "jam-free insertion" prompts) facilitates precise peg-hole alignment, generalizing across geometric shapes as the adaptive controller compensates for depth variations in the configurations.

IV. Conclusion and Future Work

In this paper, we presented a novel framework for precise robotic peg-in-hole assembly that integrates the semantic reasoning capabilities of large language models (LLMs) with adaptive stereo visual servoing, eliminating the need for camera calibration, prior object models, or task-specific training. By leveraging LLMs to extract and correspond task-relevant geometric features from uncalibrated stereo images, our approach bridges the gap between high-level semantic understanding and low-level precision control. The semantic feature selection, guided by natural language prompts, ensures robust identification of alignment points that satisfy geometric and contextual constraints for collision-free insertion. Coupled with an adaptive controller that online estimates intrinsic and extrinsic parameters, the system achieves sub-pixel accuracy in feature alignment, enabling successful peg-in-hole operations with clearances under 3 mm. Our extensive experiments on cylindrical, square, and hexagonal peg-hole pairs across three trials demonstrated the framework's efficacy and generality. The adaptive controller converged to steady-state errors of 1.8–2.8 pixels, closely rivaling calibrated baselines (1.2–2.5 pixels), with average success rates exceeding 90% over 90 repetitions. These results underscore the framework's robustness to geometric variations and projective uncertainties, marking a significant advancement toward flexible, calibration-free visual

TABLE I: Comparison of Average Steady State Error (SSE) between Adaptive and traditional IBVS Controllers for Cylindrical peg-in-hole task

Trial	Calibrated camera parameters				Adaptive camera parameters			
	Left SSE (px)	Right SSE (px)	Left % Error	Right % Error	Left SSE (px)	Right SSE (px)	Left % Error	Right % Error
1	1.2	1.3	0.3%	0.4%	2.5	1.8	0.5%	0.6%
2	1.4	2.5	0.4%	0.6%	2.2	2.7	0.6%	0.6%
3	1.7	1.5	0.5%	0.7%	2.5	2.8	0.7%	1.3%

TABLE II: Comparison of Average Success Rate for Different Object Pairs

Trial	Average Success Rate			
	Cylindrical Pairs	Square Pairs	Hexagonal Pairs	Mix Pairs
1	100%	100%	90%	90%
2	100%	90%	90%	90%
3	90%	90%	90%	90%

TABLE III: Comparison of Average Steady State Error (SSE) between Adaptive and traditional IBVS Controllers

Trial	Traditional		Adaptive	
	Left SSE (px)	Right SSE (px)	Left SSE (px)	Right SSE (px)
1	1.2	1.3	2.5	1.8
2	1.4	2.5	2.2	2.7
3	1.7	1.5	2.5	2.8

servoing in unstructured environments. By combining the interpretability of classical control with the zero-shot generalization of LLMs, our method paves the way for more autonomous and adaptable robotic manipulation systems.

A. Future Work

While our framework demonstrates strong performance in peg-in-hole tasks, future work will focus on three key extensions to broaden its applicability. First, we plan to extract higher-dimensional features, such as lines or contours, using LLM-guided prompts and design corresponding adaptive control laws for these features. This will enable robust peg-in-hole insertion under complex conditions, including irregular terrains or non-standard shapes. Second, integrating force-torque feedback into the control law will enhance robustness during contact-rich phases, allowing for more reliable and efficient operations in unstructured environments. Third, we will employ more keypoint pairs to achieve full 6-DoF visual servoing and evaluate the accuracy in the operational space.

APPENDIX

LLM FEATURE EXTRACTION PROMPT SAMPLE

This is the extracted surface feature points from the image. Now we want to perform a peg-in-hole insertion task. Please filter, merge, and extract effective image feature points that represent the task process from the original points, satisfying the following requirements:

1) Demonstrate the distinction between the hollow cylinder and the outer shell relevant to the hole insertion task;
2) Include geometric symmetry axes, endpoints of ellipse major/minor axes, and edge endpoints;
3) The cylinder central axis should have points distributed uniformly on both sides, with all points' distances to the central axis close to the median of all distance sums;
4) Focus primarily on inner edge features and the central axis geometry of the peg/hole, excluding surface features;
5) Ensure these are consistently detectable, precise, and valid expressions;
6) Output in two parts: one with no more than 8 points, and another with 4 core points selected from the original extraction.

Output example:

```
# Example output structure as specified
(peg or hole)_feature_points = {
    "xxx": (500, 300)...}
(peg or hole)_core_points = {
    "xxx": (500, 300)...}
```

Hole feature points (AKAZE): u=363.00, v=128.00 u=314.00, v=140.00 u=357.00, v=152.00 u=303.00, v=160.00 u=380.00, v=161.00 u=319.00, v=162.00 u=385.00, v=179.00 u=307.00, v=197.00 u=383.00, v=204.00 u=584.00, v=221.00 u=403.00, v=222.00 u=302.00, v=226.00 u=291.00, v=250.00 u=409.00, v=252.00 u=394.00, v=269.00 u=305.00, v=274.00 u=378.00, v=278.00 u=321.00, v=278.00 u=361.00, v=285.00 u=334.00, v=128.00 u=332.00, v=153.00 u=345.00, v=166.00 u=405.00, v=237.00 u=367.00, v=184.00 u=390.00, v=250.00 u=312.00, v=254.00 u=352.00, v=253.00 u=350.00, v=238.00
LLM results:

```
hole_feature_points = {
    "top_left": (303.00, 160.00),
    "top_right": (380.00, 161.00),
    "bottom_left": (291.00, 250.00),
    "bottom_right": (409.00, 252.00),
    "left_mid": (307.00, 197.00),
    "right_mid": (385.00, 179.00),
    "center_top": (357.00, 152.00),
    "center_bottom": (361.00, 285.00)}
hole_core_points = {
    "top_center": (357.00, 152.00),
```

```
    "bottom_center": (361.00, 285.00),
    "left_center": (303.00, 226.00),
    "right_center": (394.00, 237.00)}
```

The task is mainly divided into two phases: approach and alignment. The first phase prevents collision, and the second phase ensures alignment. Select one visible original feature point from each of the peg and the hole. Both points should be clearly visible and trackable, while maximizing the avoidance of collisions with other points when mutual constraints are applied. Explain the mutual relationship and constraints at this task stage, with point-to-point alignment features prioritized. Please constrain to a single point to facilitate visual servoing control. For example: In phase xxx, point xxx aligns with point xxx with uv constraint of xxx.

Output format:

```
# Example output structure as specified
peg_feature = {"xxx": (500, 300)}
hole_feature = {"xxx": (500, 300)}
relations = {
    "approach": relationship constraints,
    "alignment": relationship constraints}

peg_feature = {
    "bottom_center": (598, 449)}
hole_feature = {
    "top_center": (226, 398)}
relations = {
    "approach": "|u_peg - u_hole| < 50
        and v_peg > v_hole + 100",
    "alignment": "u_peg = u_hole
        and v_peg = v_hole + 30"}
```

REFERENCES

[1] L. Shen, J. Su, and X. Zhang, "Review on peg-in-hole insertion technology based on reinforcement learning," in *2023 China Automation Congress (CAC)*, 2023, pp. 6688–6695.

[2] K. Van Wyk, *et al.*, "Comparative peg-in-hole testing of a force-based manipulation controlled robotic hand," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 542–549, 2018.

[3] K. Zhang, *et al.*, "Vision-based six-dimensional peg-in-hole for practical connector insertion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1771–1777.

[4] Z. Deng and M. Jagersand, "Evaluation of model independent image-based visual servoing," in *First Canadian Conference on Computer and Robot Vision, 2004. Proceedings.*, 2004, pp. 138–144.

[5] P. Pashupathy, *et al.*, "Force-based online estimation and adaptive alignment for robotic peg-in-hole insertion," in *2025 11th International Conference on Control, Automation and Robotics (ICCAR)*, 2025, pp. 151–156.

[6] Q. Meng, J. Gu, and Y.-H. Liu, "Gpd: Learning geometric primitive deformation for unseen object pose estimation," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 9903–9922, 2025.

[7] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[8] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 d visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999.

[9] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[10] B. Chen, *et al.*, "Open-vocabulary queryable scene representations for real world planning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 509–11 522.

[11] O. Mees, J. Borja-Diaz, and W. Burgard, "Grounding language with visual affordances over unstructured data," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 576–11 582.

[12] B.-S. Lu, *et al.*, "Cfvs: Coarse-to-fine visual servoing for 6-dof object-agnostic peg-in-hole assembly," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 402–12 408.

[13] S. Levine, *et al.*, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.

[14] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, 2016.

[15] A. Rajeswaran, *et al.*, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv preprint arXiv:1709.10087*, 2017.

[16] Y. Qin, "Learning generalizable dexterous manipulation," Ph.D. dissertation, University of California, San Diego, 2024.

[17] C. Chi, *et al.*, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, p. 02783649241273668, 2023.

[18] X. Chen, *et al.*, "Adversarial feature training for generalizable robotic visuomotor control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1142–1148.

[19] C. Yu, *et al.*, "Siamese convolutional neural network for sub-millimeter-accurate camera pose estimation and visual servoing," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 935–941.

[20] P. Zhang, *et al.*, "Vinvl: Revisiting visual representations in vision-language models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5579–5588.

[21] M. Ahn, *et al.*, "Do as i can and not as i say: Grounding language in robotic affordances," in *arXiv preprint arXiv:2204.01691*, 2022, preprint.

[22] W. Huang, *et al.*, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[23] J. Hu, *et al.*, "Precision motion control of a 6-dofs industrial robot with accurate payload estimation," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 4, pp. 1821–1829, 2020.

[24] ——, "Desired compensation adaptive robust repetitive control of a multi-dofs industrial robot," *ISA Transactions*, vol. 128, pp. 556–564, 2022.

[25] G. Sun, *et al.*, "Smooth surface-to-surface contact control for rope-base soft-tip manipulator," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 16 297–16 308, 2025.

[26] C.-C. Cheah, C. Liu, and J.-J. E. Slotine, "Adaptive jacobian tracking control of robots based on visual task-space information," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 3498–3503.

[27] Y.-H. Liu, *et al.*, "Uncalibrated visual servoing of robots using a depth-independent interaction matrix," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 804–817, 2006.

[28] H. Wang, *et al.*, "Adaptive visual servoing of contour features," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 811–822, 2018.

[29] F. Xu, *et al.*, "Adaptive visual servoing shape control of a soft robot manipulator using bézier curve features," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 2, pp. 945–955, 2023.

[30] Y. Zhang, *et al.*, "Adaptive Visual Servoing Control of Closed-Architecture Robots With Both Visibility Constraints and Tracking Error Constraints," *IEEE/ASME Transactions on Mechatronics*, pp. 1–10, 2025.

[31] J.-J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, 1991.

[32] A. Jokic, M. Petrovic, and Z. Miljkovic, "Semantic segmentation based stereo visual servoing of nonholonomic mobile robot in intelligent manufacturing environment," *Expert Systems with Applications*, vol. 190, p. 116203, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421015189

[33] S. Auddy, *et al.*, "Imitation learning-based direct visual servoing using the large projection formulation," *Robotics and Autonomous Systems*, vol. 190, p. 104971, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889025000570