PPO-IBVS:A Residual RL Framework for Smooth and Energy-Aware Visual Servoing

Qihao Qian
ECE department
UC San Diego
q2qian@ucsd.edu

***Abstract*—Image-Based Visual Servoing (IBVS) is a widely used control strategy for robotic manipulation due to its robustness against calibration errors. However, classical IBVS controllers rely on greedy optimization of geometric error, often resulting in energetically inefficient trajectories and erratic joint velocities. This paper presents a Residual Reinforcement Learning (RL) framework to address these limitations. We propose a hybrid control architecture where a Proximal Policy Optimization (PPO) agent learns a residual modulation over a nominal IBVS controller. By observing an augmented state space that includes visual features, proprioceptive data, and the nominal baseline suggestion, the agent learns to optimize for energy efficiency and motion smoothness without sacrificing convergence accuracy. Experimental results in a physics-based simulation demonstrate that our method reduces energy consumption by 56.68% and improves overall tracking rewards by 11.57% compared to the standard analytical baseline.**

I. Introduction

Visual servoing has long been a cornerstone of robotic manipulation, enabling systems to interact with dynamic targets in unstructured environments by closing the control loop directly in the sensor space [1]. Among these techniques, Image-Based Visual Servoing (IBVS) is particularly favored for its robustness to calibration errors and low computational overhead, as it avoids explicit 3D reconstruction. However, as robotic applications expand into energy-constrained and human-centric domains, the requirements for control policies have evolved beyond simple geometric convergence. Modern systems demand trajectories that are not only accurate but also energy-efficient and kinematically smooth.

Classical IBVS controllers, typically formulated as gradient descent on image plane errors, are inherently “greedy.” They optimize for the shortest path in the 2D image feature space without regard for the robot’s physical configuration or dynamics. As demonstrated in our baseline implementation, the classical controller calculates camera velocities purely to minimize pixel error. While this ensures geometric convergence, it treats system dynamics—such as joint velocity limits, torque consumption, and motion smoothness—as secondary constraints rather than optimization objectives. Consequently, standard IBVS often produces trajectories that are geometrically valid but physically suboptimal, characterized by erratic velocity profiles or excessive energy consumption.

To address these limitations, Reinforcement Learning (RL) offers a promising alternative by allowing the optimization of complex, non-differentiable reward functions. However, end-to-end RL approaches often suffer from poor sample efficiency and lack the stability guarantees of classical control theory.

In this work, we propose a Residual Reinforcement Learning framework for visual servoing that combines the geometric robustness of IBVS with the optimization versatility of Deep RL. Rather than learning a control policy from scratch, our approach utilizes a classical IBVS controller to generate a nominal policy. A Proximal Policy Optimization (PPO) agent then learns a residual modification to this nominal command [2].

This hybrid architecture allows the system to leverage the baseline controller for initial guidance while the RL agent fine-tunes the trajectory to satisfy auxiliary objectives that the classical controller ignores. Specifically, we introduce a reward structure that penalizes instantaneous power consumption (approximated by the product of torque and velocity) and maximizes motion smoothness.

The primary contributions of this paper are as follows:

- **A Residual Control Architecture:** We present a formulation where an RL policy modulates the camera velocity commands ($\mathbf{v}_{cam}$) generated by an analytical IBVS controller, maintaining geometric convergence while improving dynamic performance.
- **Energy and Smoothness Optimization:** We demonstrate that by incorporating energy and smoothness terms into the reward function—factors typically absent in the Jacobian-based inversion of the baseline—the system can achieve successful servoing with significantly reduced mechanical cost.
- **State-Aware Policy Learning:** We introduce an augmented observation space that fuses visual feedback (pixel error and depth) with the robot’s proprioceptive state (joint angles and velocities) and the nominal IBVS suggestion, enabling the agent to make informed decisions about when to deviate from the greedy geometric path.

II. Background

A. Camera Model and Projection

The visual servoing system relies on a pinhole camera model to map 3D world coordinates to the 2D image plane. In our simulation, the camera is mounted in an eye-in-hand

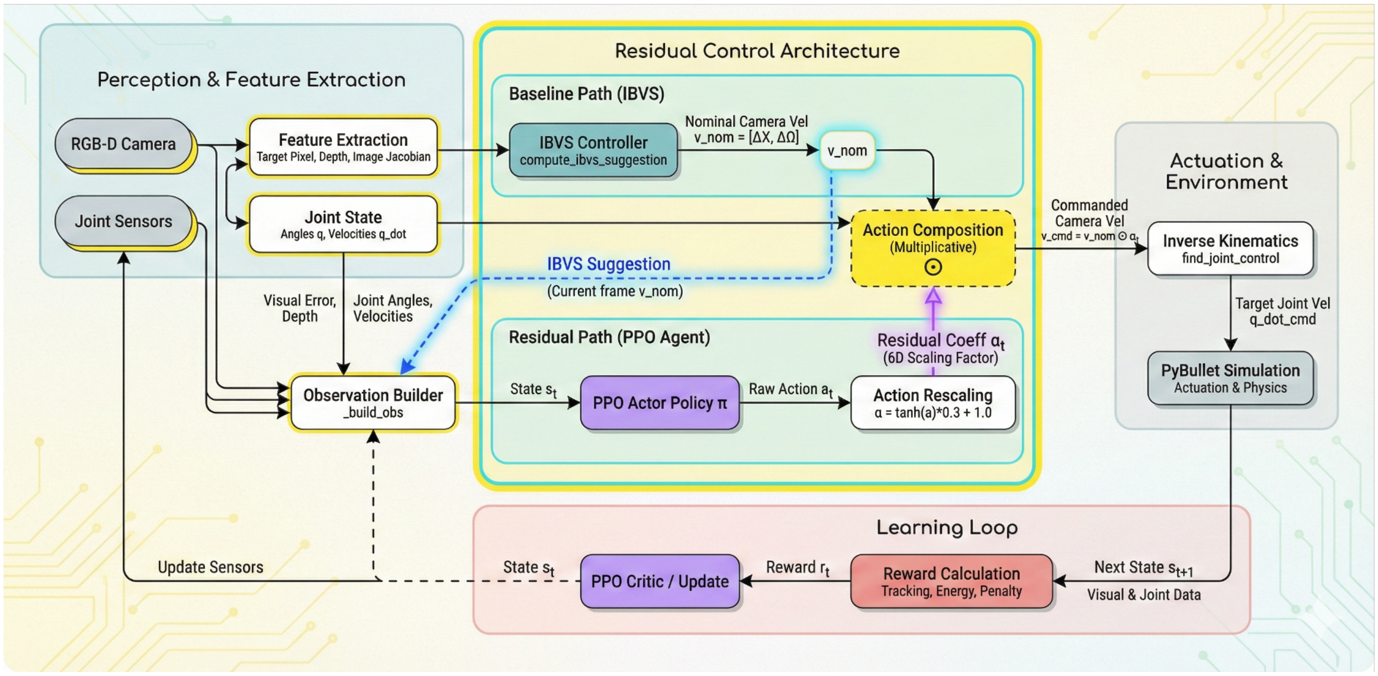

Fig. 1. **Residual Control Pipeline.** The system operates on two parallel paths: the *Baseline Path* (top cyan block) computes the nominal IBVS velocity $\mathbf{v}_{nom}$ based on geometric features. The *Residual Path* (bottom teal block) utilizes a PPO agent to observe the system state—including the baseline's suggestion—and outputs a scaling coefficient α_t. These are fused in the *Action Composition* block to generate the final commanded velocity $\mathbf{v}_{cmd}$, effectively allowing the agent to "throttle" or "boost" the analytical controller to minimize energy consumption ($\sum \dot{q}^2$) and tracking error.

configuration on the robot's end-effector. Let $\mathbf{P} = [X, Y, Z]^\top$ denote a point in the camera reference frame. Its projection onto the image plane, denoted by pixel coordinates $\mathbf{s} = [u, v]^\top$, is given by:

$$u = u_0 + f\frac{X}{Z}, \quad v = v_0 + f\frac{Y}{Z} \tag{1}$$

where (u_0, v_0) is the principal point (the image center) and f is the focal length in pixel units. This projection is implemented in our system via the opengl plot world to pixelspace function, which transforms world coordinates using the camera's view and projection matrices.

B. Image-Based Visual Servoing (IBVS)

The baseline controller relies on the Interaction Matrix (or Image Jacobian), L_s, which linearizes the relationship between feature motion in the image plane and camera velocity in 3D space.

For a point feature $\mathbf{s} = [u, v]^\top$, the time derivative $\dot{\mathbf{s}}$ is related to the camera's spatial velocity $\mathbf{v}_c$ by $\dot{\mathbf{s}} = L_s(\mathbf{s}, Z)\mathbf{v}_c$. Our implementation utilizes the analytical form of this matrix for normalized coordinates:

$$L_s = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix} \tag{2}$$

The baseline control law minimizes the error $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ by inverting this relationship:

$$\mathbf{v}_c = -\lambda \widehat{L_s}^+ \mathbf{e} \tag{3}$$

where λ is a gain term and $\widehat{L_s}^+$ is the pseudoinverse of the Jacobian. While effective for geometric convergence, this method does not optimize for dynamic constraints like energy or smoothness.

C. Reinforcement Learning and PPO

To overcome the limitations of the analytical baseline, we model the control problem as a Markov Decision Process (MDP) and solve it using Reinforcement Learning (RL). In this framework, an agent interacts with the environment by observing a state s_t, executing an action a_t, and receiving a reward r_t. The goal is to learn a policy $\pi_\theta(a_t|s_t)$ that maximizes the expected cumulative reward.

We specifically utilize Proximal Policy Optimization (PPO), a policy gradient method known for its stability and ease of tuning. PPO avoids large, destructive policy updates by optimizing a "surrogate" objective function that clips the probability ratio between the new and old policies:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{4}$$

where $r_t(\theta)$ is the probability ratio $\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$, $\hat{A}_t$ is the estimated advantage, and ϵ is a hyperparameter determining the clipping range (typically 0.2).

In our residual framework, PPO does not output the raw motor commands directly. Instead, it outputs a residual coefficient that modulates the nominal velocity suggested by the IBVS controller. This allows the PPO agent to focus on learning high-level corrections (e.g., smoothing motion or reducing torque) without needing to re-learn the fundamental geometry of visual servoing.

III. Method

In this work, we propose a Residual Reinforcement Learning framework that augments a classical Image-Based Visual Servoing (IBVS) controller with a learned residual policy. This hybrid architecture, illustrated in Figure 1, leverages the geometric stability of the analytical baseline while enabling the optimization of secondary dynamic objectives such as energy efficiency and motion smoothness.

A. Residual Control Architecture

Unlike end-to-end approaches that map pixels directly to joint torques, our method learns a modulation over a nominal kinematic controller. As shown in the "Residual Control Architecture" block of Figure 1, the control law is defined as a multiplicative composition:

$$\mathbf{v}_{cmd} = \mathbf{v}_{nom} \odot \alpha_t \tag{5}$$

where $\mathbf{v}_{nom} \in \mathbb{R}^6$ is the nominal camera velocity computed by the classical IBVS controller (see Section II-B), and $\alpha_t \in \mathbb{R}^6$ is the residual action vector generated by the RL policy. This structure ensures that if the agent outputs $\alpha_t \approx \mathbf{1}$, the system defaults to the stable baseline behavior.

B. State Space and Observation Builder

A critical innovation in our pipeline is the *Observation Builder*, which fuses visual perception with proprioception and, uniquely, the *intent* of the baseline controller. The state vector $s_t \in \mathbb{R}^{23}$ is constructed as follows:

- **Visual Error ($\mathbb{R}^2$):** The normalized vector from the image center to the target pixel (u, v).
- **Depth ($\mathbb{R}^1$):** The depth estimate Z at the target centroid, which scales the interaction matrix.
- **Proprioception ($\mathbb{R}^{14}$):** The robot's current joint angles $\mathbf{q}$ and joint velocities $\dot{\mathbf{q}}$.
- **IBVS Suggestion ($\mathbb{R}^6$):** The nominal velocity $\mathbf{v}_{nom}$ that the baseline controller *would* execute in the current frame.

By including the "IBVS Suggestion" explicitly in the observation, the PPO agent does not need to learn the geometry of visual servoing from scratch; instead, it learns how the baseline's proposed action correlates with energy costs and future rewards.

C. Action Space and Modulation

The PPO agent outputs a raw action vector $a_t \in \mathbb{R}^6$, which corresponds to the 6 degrees of freedom of the camera (3 translational, 3 rotational). To ensure stability, this raw output is passed through a *Rescaling* function (seen in Figure 1) to produce the modulation coefficient α_t:

$$\alpha_t = 0.3 \cdot \tanh(a_t) + 1.0 \tag{6}$$

This bounds the modulation $\alpha_t \in [0.7, 1.3]$, effectively limiting the agent to deviating by at most 30% from the baseline's geometric solution. This constraint prevents the agent from learning unsafe or divergent behaviors during early training phases.

D. Reward Function

The agent is trained to maximize a composite reward function R_t that balances tracking accuracy against mechanical effort:

$$R_t = r_{track} + r_{energy} + r_{penalty} + r_{fail} \tag{7}$$

- **Tracking Reward (r_{track}):** Penalizes the Euclidean distance between the target pixel and the image center.
- **Energy Cost (r_{energy}):** Penalizes high-velocity movements to encourage efficiency, approximated by the sum of squared joint velocities: $r_{energy} = -w_e \sum \dot{q}^2$.
- **Step Penalty ($r_{penalty}$):** A constant negative reward per time step to encourage faster convergence.
- **Failure Penalty (r_{fail}):** A large penalty if the target leaves the camera's field of view.

As implemented in our simulation environment, the weights are set to $w_{track} = 1.0$, $w_{steppenalty} = 10.0$, $w_{energy} = 100.0$, and $w_{fail} = 1000.0$, prioritizing safety and visibility while strictly optimizing for energy economy.

IV. Experiments and Results

To validate the efficacy of the proposed Residual RL framework, we conducted a comparative evaluation against the classical IBVS baseline. Both controllers were tested over a series of randomized episodes in the PyBullet simulation environment. The primary metrics for evaluation correspond to the components of our reward function: tracking accuracy, energy consumption (approximated by joint velocity squared), and task success rate.

A. Quantitative Results

The aggregate results, averaged over the evaluation episodes, are presented in Table I. The metrics are reported as negative values (rewards/costs), where values closer to zero indicate better performance.

B. Discussion and Analysis

The results demonstrate that the Residual PPO-IBVS controller significantly outperforms the analytical baseline across all metrics, achieving a **45.31% improvement** in total cumulative reward.

1) Energy Efficiency: The most profound impact of the residual policy is observed in the *Energy Reward*, which improved by **56.68%** (from -13605.69 to -5893.71). The classical baseline, driven by a greedy Jacobian inversion, often commands high joint velocities to minimize pixel error instantly. In contrast, the RL agent learns to "throttle" these commands. By observing the IBVS Suggestion alongside the robot's current joint velocities, the agent identifies when the baseline's request is excessively costly and outputs a scaling

TABLE I
PERFORMANCE COMPARISON: BASELINE IBVS VS RESIDUAL PPO-IBVS

Metric	Baseline (Avg)	PPO-IBVS (Avg)	Abs. Reduction	Improv. (%)
Cumulative Reward	$-18\,029.954$	-9859.780	8170.174	45.310
Tracking Reward	-3217.465	-2845.343	372.122	11.570
Energy Reward	$-13\,605.689$	-5893.710	7711.979	56.680
Step Penalty	-956.800	-933.000	23.800	2.490
Failure Reward	-250.000	-190.000	60.000	24.000

coefficient $\alpha_t < 1$, effectively smoothing the trajectory without sacrificing convergence.

2) Tracking and Robustness: Contrary to the typical trade-off between energy efficiency and accuracy, the residual controller also improved the *Tracking Reward* by **11.57%**. This suggests that the baseline's aggressive movements often lead to overshooting or unstable behavior that hampers precise convergence. By damping these oscillations, the residual policy achieves a steadier approach to the target.

Furthermore, the *Failure Reward* improved by **24.00%**, indicating that the RL agent is more robust against losing the target from the camera's field of view. The baseline blindly follows the gradient, which can lead to singular configurations or erratic camera motions that cause target loss. The PPO agent, incentivized by the heavy failure reward penalty, learns to modify the trajectory to keep the target safely within the frame, even if it requires a slight deviation from the shortest path.

V. CONCLUSION

In this work, we developed and evaluated a Residual Reinforcement Learning framework for visual servoing that successfully bridges the gap between classical geometric control and learning-based optimization. By structuring the control policy as a multiplicative modulation of a nominal IBVS law, we retained the geometric guidance of the Jacobian-based controller while empowering an RL agent to mitigate its inherent inefficiencies.

Our experiments confirmed that the baseline controller, while effective at minimizing pixel error, operates with a ”greedy” strategy that neglects the robot's dynamic constraints. The proposed residual agent successfully learned to identify and dampen these inefficient commands. As evidenced by a **45.31% increase in cumulative reward**, the system achieved a superior balance between tracking accuracy and mechanical effort. The significant reduction in energy cost (**56.68%**) validates that residual learning is a potent strategy for refining classical control algorithms, offering a practical path toward deployment on physical robots where energy conservation and smooth actuation are critical. Future work will focus on transferring this policy to a physical Franka Emika Panda robot to validate its performance under real-world noise and latency conditions.

REFERENCES

[1] D. Kragic and H. I. Christensen, “Survey on visual servoing for manipulation,” *Computational Vision and Active Perception Laboratory, Fiskartorpsv*, vol. 15, pp. 2002, 2002

[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.