Robotic Sim-to-Real Transfer for Long-Horizon Pick-and-Place Tasks in the Robotic Sim2Real Competition

Ming Yang[1,2*], Hongyu Cao[3*], Lixuan Zhao[3], Chenrui Zhang[3], Yaran Chen[1,2✉]

***Abstract*— This paper presents a fully autonomous robotic system that performs sim-to-real transfer in complex long-horizon tasks involving navigation, recognition, grasping, and stacking in an environment with multiple obstacles.**

The key feature of the system is the ability to overcome typical sensing and actuation discrepancies during sim-to-real transfer and to achieve consistent performance without any algorithmic modifications. To accomplish this, a lightweight noise-resistant visual perception system and a nonlinearity-robust servo system are adopted.

We conduct a series of tests in both simulated and real-world environments. The visual perception system achieves the speed of 11 ms per frame due to its lightweight nature, and the servo system achieves sub-centimeter accuracy with the proposed controller. Both exhibit high consistency during sim-to-real transfer.

Our robotic system took first place in the mineral searching task of the Robotic Sim2Real Challenge hosted at ICRA 2024. The simulator is available from the competition committee at `https://github.com/AIR-DISCOVER/ICRA2024-Sim2Real-RM`, and all code and competition videos can be accessed via our GitHub repository at `https://github.com/Bob-Eric/rmus2024_solution_ZeroBug`.

I. Introduction

The pick-and-place task, which integrates key functionalities such as perception and servo control, is a fundamental skill of embodied intelligence. It is critical for a wide range of applications, from autonomous sorting in a warehouse to item organizing by household robots. However, algorithms developed in simulators often face significant discrepancies when transferred to real-world robotic systems, leading to severe performance degradation or even total failure. This challenge is commonly referred to as the sim-to-real gap.

Traditional approaches to addressing the sim-to-real gap typically involve applying reinforcement learning algorithms in simulators [29], followed by transferring them to real systems with the aid of domain randomization [1, 17], transfer learning [2, 3, 16] in real-world settings. However, the noise introduced by domain randomization is often insufficient for generalization to complex environments, and transfer learning in the real-world often requires extensive interaction data [22], which is difficult to obtain.

[1]The State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, {yangming2023, chenyaran2013}@ia.ac.cn

[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China.

[3]School of Electrical and Information Engineering, Tianjin University, Tianjin, China.

∗Equal contribution.

✉Corresponding to chenyaran2013@ia.ac.cn

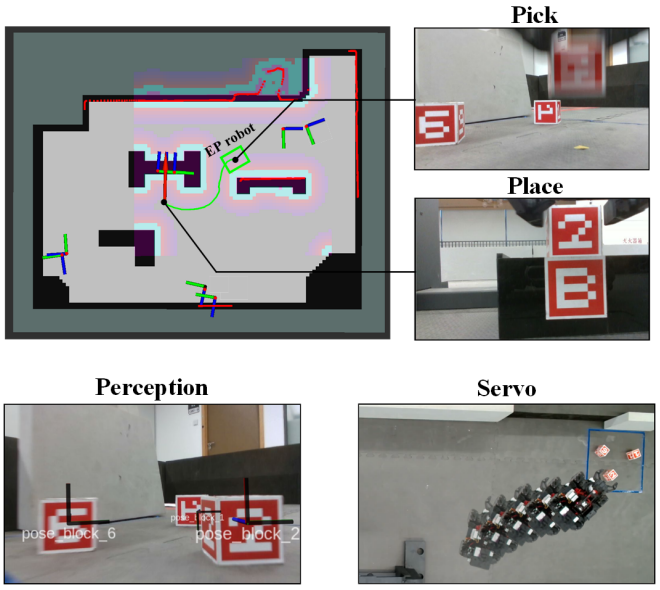


Fig. 1. Our robot navigating to the exchange station to place a mineral after grasping it. High-precision visual perception (a) and accurate servo control (b) are critical for successful picking and placing.

Therefore, the main propose of this paper is to demonstrate that **it is both possible and feasible to achieve consistent performance in long-horizon pick-and-place tasks across simulated and real-world environments without any modification to the algorithm.**

To make this possible, our robotic system consists of two components, with the aim of addressing sensing and actuation discrepancies, respectively. The first is a visual perception system that operates on a detection-classification-localization pipeline with proposed Sequential Motion-blur Mitigation Strategy (SMMS) to minimize the impact of the sensing discrepancy throughout the pipeline. The second is a feedback-linearized servo system with proposed Design Function (DF), which addresses actuation discrepancies of both inappropriate grasp poses (illustrated in Fig. 5) and unmodeled nonlinearities.

These algorithms were developed as part of the ZeroBug Team system that took first place in the mineral searching task in the Robotic Sim2Real Challenge (RSC) [30]. Fig. 1 shows our robot in action during the simulation and real-world stages of the competition.

The main contributions of this paper are summarized as follows.

1) **High-Consistency Visual Perception System**: We design Sequential Motion-Blur Mitigation Strategy for our visual perception system to handle typical sensing errors, delivering consistent performance in both simulated and real-world environments.
2) **Nonlinearity-Robust Servo System**: We introduce Design Function to our feedback-linearized servo system to mitigate actuation discrepancies, demonstrating strong robustness to nonlinearities.
3) **Modular System Architecture**: The robotic system features a modular architecture, offering a flexible and adaptable platform for researchers.

II. Related Works

In recent years, a variety of simulators have been introduced for embodied intelligence [9, 13, 19, 20, 25, 26]. In these simulators, sim-to-real gap can be broadly categorized into sensing discrepancies (variations in color and texture, non-ideal imaging, etc.) and actuation discrepancies (modeling error, nonlinearities, condition-based actuations, etc.) [29].

For sensing discrepancies, since Tobin et al. [21] first introduce domain randomization, it is widely used in simple and stationary environments [8, 23, 27]. However, for mobile manipulators with on-board cameras, motion blur, which domain randomization fall short in addressing, may be one of the most common challenges (illustrated in Fig. 2). In section III, we propose SMMS to minimize the impact of motion blur.

(a) (b)

Fig. 2. Images of markers captured in the simulator (a) and real-world (b) while robot rotating. Motion blur is difficult to simulate and poses challenges to robotic perception.

For actuation discrepancies, popular approaches include reinforcement learning with fine-tuning [4, 12, 28], dynamics randomization [1, 17] and domain adaptation [2, 3, 16], all of which require extensive interactions. In section IV, we introduce DF to adapt to reality with minimal interactions.

The 2024 RSC simulator, with the task of searching minerals scattered around the scene and stacking them in the exchange station as high as possible, is built with Habitat [19]. However, given that the physics and rendering engines used in mainstream simulators are similar, we believe proposed methods for overcoming sensing and actuation discrepancies can be broadly applicable to other simulators.

III. Challenge I: Perceiving with Sensing Discrepancies

The first step in our system is to achieve consistently precise and real-time pose estimation. Learning-based algorithms [7, 10, 18], while offering high accuracy and strong robustness in detecting and localizing fiducial markers, often suffer from high computational overhead and extended runtimes. In contrast, traditional computer vision techniques (ArUco [6, 11], AprilTag [15, 24]) are inherently more efficient but struggle with robustness, particularly in handling noise or image defects. To balance accuracy and efficiency, we replace the template matching approach in the ArUco detector with a CNN-based classifier (illustrated in Fig. 3), drawing on a network architecture similar to LeNet-5 [14].

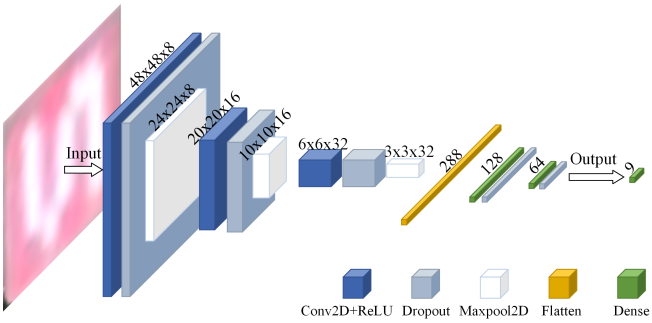


Fig. 3. The CNN classifier comprises three convolution-dropout-pooling modules, followed by three fully connected layers. Its lightweight architecture (62.1k parameters in total) enables real-time inference.

Additionally, motion blur, as the most common sensing discrepancy during sim-to-real transfer, poses great challenges to perceptual performance. It affects pose estimation of each mineral through the following path: motion blur leads to false recognitions and deformed contours, both of which cause imperfect images of markers after perspective transformation, and thereby misclassifications. Subsequently, deformed or misclassified contours introduce noise or abrupt change to pose estimates. To address this, we propose Sequential Motion-blur Mitigation Strategy (SMMS), illustrated in Fig. 4.

Image Enhancement. Vanilla ArUco detector [6] suffers great loss of recall rate due to motion blur, for which we boost image contrast to compensate. Specifically, we first split RGB channels of the given image and boost the red channel (R) with the difference of R and B. Then we subtract G from the boosted R to get the feature map (AKA, the input of ArUco detector):

$$feature = R + k \cdot max(R - B, 0) - G \tag{1}$$

where $k \in (0, +\infty)$ is an enhancement factor and set to 1.5 here. Compared with weighted sum of R, G and B adopted by vanilla ArUco detector ($feature = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$), it sharpens the boundary of red mineral markers, thereby raising the recall rate from 50.5% to 70.0% in real-world (see Table I).

Data Augmentation. We first collect 5 to 9 samples for each class, and then augment the training set with random rotation and color jitter to adapt to reality. Specifically, we first apply random rotations between ± 10 degrees to adapt to deformation and skew of transformed images. Then we apply random perturbations to the brightness, contrast, saturation, and hue of each training image.

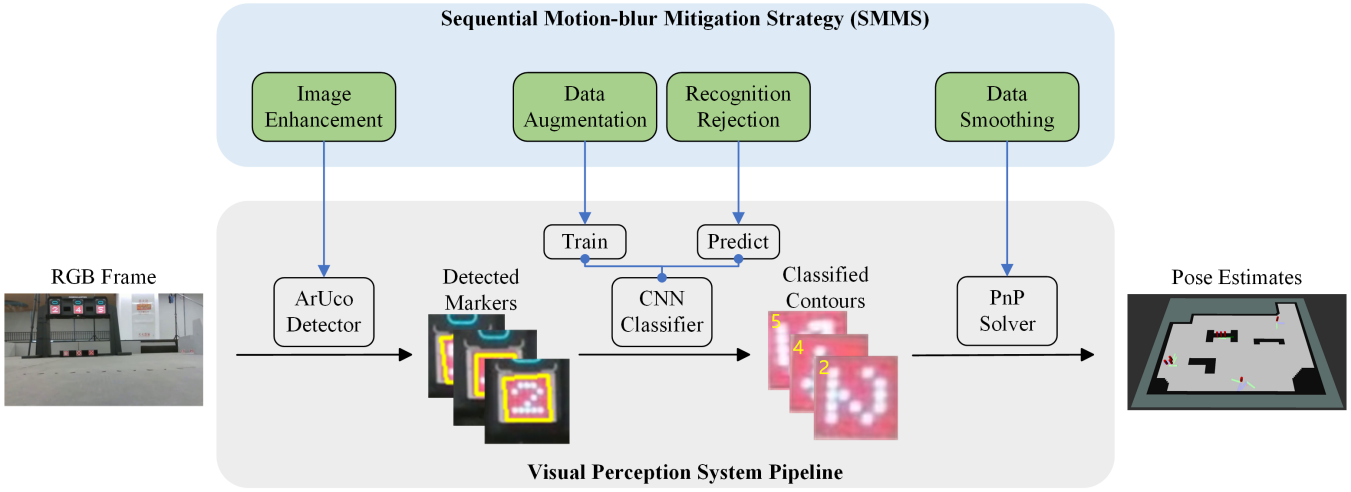


Fig. 4. Pipeline of our visual perception system with SMMS. We first use ArUco for extracting contours of markers. Then we rectify each contour with perspective transformation and classify it with CNN classifier, followed by a Perspective-n-Points (PnP) solver to estimate its pose. Throughout the pipeline, SMMS first enhances the image contrast before feeding the feature map into the detector. It then augments training dataset and rejects unreliable classifications. Subsequently, it applies extrapolation and filtering for final pose estimates.

Recognition Rejection. Incorrect contours produced by the ArUco detector are assigned labels based on the highest score of nine categories, leading to erroneous pose estimation of the corresponding minerals. To address this, we analyze the numerical features of incorrect contours and establish a threshold-based rejection rule. Briefly, it eliminates false recognitions entirely with minimal loss of detector's recall rate.

Data Smoothing. To further mitigate the impact of motion blur, we apply extrapolation for undetected minerals and low-pass filter for the detected ones, which compensate for loss of recall rate and precision, respectively. Concretely, we hold the global pose of undetected minerals and transform it to the local pose for closed-loop servo control, and we filter global poses of detected minerals to eliminate measurement noise introduced by motion blur as follows:

$$p_n = a \cdot p_{n-1} + (1-a) \cdot p_n^{obs} \tag{2}$$

where p_n and p_{n-1} are estimated mineral pose at frame n and $n-1$, p_n^{obs} is the raw output of the PnP solver at frame n, and $a \in [0, 1]$ is a filter coefficient.

In summary, SMMS mitigates the impact of motion blur as follows: image enhancement compensates for contrast loss, improving the recall rate; data augmentation boosts the classifier's generalization ability, and the rejection mechanism eliminates false recognitions, maintaining classification accuracy; data smoothing ensures pose estimation precision and smoothness. Through SMMS, we successfully bridge the sim-to-real gap of sensing discrepancies in the 2024 RSC.

IV. Challenge II: Servo with Actuation Discrepancies

With real-time pose estimates available, the second step for the robotic system is to robustly and precisely align the gripper with the target mineral. However, there are two main actuation discrepancies in mainstream simulators. The first is inappropriate grasp poses, as illustrated in Fig. 5, which may be the most notorious cause of inconsistent grasp results during sim-to-real transfer [9, 13, 19]. The second is the nonlinearity in actual dynamics, in which mainstream simulators fall short [13, 20]. To address both, we first integrate angular alignment into chassis servo control with feedback linearization. Then we introduce Design Function (DF) to fulfill expected attenuation mode, thereby achieving necessary robustness.

(a) (b)

Fig. 5. Incorrect grasp results caused by inappropriate grasp pose. When grasping the mineral diagonally in practice (a), the gripper tends to get stuck on the corner and fail. However, it "penetrates" into the mineral and succeeds in the simulator (b).

A. Modeling Chassis

Let the pose of the target in the chassis coordinate system be $(x, y, \theta)^T$, and the velocity input to the omnidirectional chassis be $(v_x, v_y, \omega)^T$. As illustrated in Fig. 6, positional and angular changes in marginal time increment δt induced by v_x, v_y, ω are

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos\delta\theta & \sin\delta\theta \\ -\sin\delta\theta & \cos\delta\theta \end{pmatrix} \begin{pmatrix} x + v_x\delta t \\ y + v_y\delta t \end{pmatrix} \tag{3}$$

and

$$\theta_2 = \theta - \omega\delta t \tag{4}$$

Then by letting $\delta t \to 0$, we obtain the derivative of state variables as follows:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} v_x - y\omega \\ v_y + x\omega \\ -\omega \end{pmatrix} \tag{5}$$

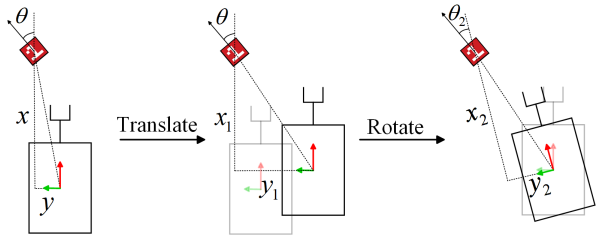


Fig. 6. Marginal pose change induced by translation and rotation. The x and y axes of the chassis coordinate system are shown in red and green, respectively.

B. Feedback Linearization

Here, we apply feedback linearization through inverse dynamics [5]. For any given trajectory $(x_d(t), y_d(t), \theta(t))^T, t \in \mathbb{R}$, the state error of the system can be denoted as $\mathbf{e} = (x - x_d, y - y_d, \theta - \theta_d)^T$ [1]. To attenuate it to 0, the derivative of the state error can be designed as follows:

$$\dot{\mathbf{e}} = \mathbf{f}(\mathbf{e}, t) \tag{6}$$

where $\mathbf{f} = (f_1, f_2, f_3)^T$ is the Design Function. Substituting (6) into (5), we conclude that, for any expected attenuation mode determined by $\mathbf{f}$, there exists control input such that

$$\begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix} = \begin{pmatrix} f_1 - yf_3 \\ f_2 + xf_3 \\ -f_3 \end{pmatrix} \tag{7}$$

Theoretically, $\mathbf{f}$ can be any known function of $\mathbf{e}$ and t. Here, we demonstrate three intuitive choices of $\mathbf{f}$: 1) multiplication of $\mathbf{e}$ (for exponential mode); 2) power function of $\mathbf{e}$ (for power-law mode); 3) linear combination of $\mathbf{e}$ and its integral (for multiple exponential modes).

C. Design Function

We solve the attenuation mode of state error under the above three DF choices and analyze their servo accuracy under dead zone, respectively.

For simplicity, we choose the three components of $\mathbf{f}$ as the same, where we only need to analyze in scalar form of (6):

$$\dot{e} = f(e, t) \tag{8}$$

where e can represent positional or angular error. Without loss of generality, we consider position control. In order for e to have an analytical solution, we can take f of the following form. (controller-specific parameters k_p, k_i and α are positive real numbers)

I. $f(e,t) = -k_p e$
We can immediately solve from (8) that $e = e(0) \cdot e^{-k_p t}$, indicating that the state error decays exponentially to 0.
let m denote the size of velocity dead zone, i.e.,

$$\Gamma(v) = \begin{cases} 0 & |v| \le m \\ v & |v| > m \end{cases} \tag{9}$$

where v is the velocity, Γ is the dead zone. Then the steady-state error e_{ss} satisfies $m = |-k_p e_{ss}|$, from which we get $e_{ss} = \frac{m}{k_p}$.
Type I controller offers a concise form and simplifies computation. However, the steady-state error is linearly dependent on the dead zone size, which renders the controller fragile.

II. $f(e,t) = -k_p e^{\alpha}, \quad 0 < \alpha < 1$
Assume $e(t) > 0$ [2]. By substituting this DF into (8) and integrating after separating the variables, we have $e = \left(e(0)^{1-\alpha} - (1-\alpha)k_p t\right)^{\frac{1}{1-\alpha}}$ or $e = 0$.
Likewise, e_{ss} satisfies $m = \left|-k_p {e_{ss}}^{\alpha}\right|$, from which we get $e_{ss} = \left(\frac{m}{k_p}\right)^{\frac{1}{\alpha}}$.
The rationale of Type II controller is to boost the velocity output when the error is small. In practice, m is around 0.05 m/s, α is taken as 2/3, and k_p is often taken as 1, which yields a much smaller e_{ss} than Type I.

III. $f(e,t) = -k_p e - k_i \int_0^t e(\tau) d\tau$
Substituting this DF into (8) and letting $\epsilon = \int_0^t e(\tau) d\tau$, we obtain a second-order differential equation:

$$\ddot{\epsilon} + k_p \dot{\epsilon} + k_i \epsilon = 0 \tag{10}$$

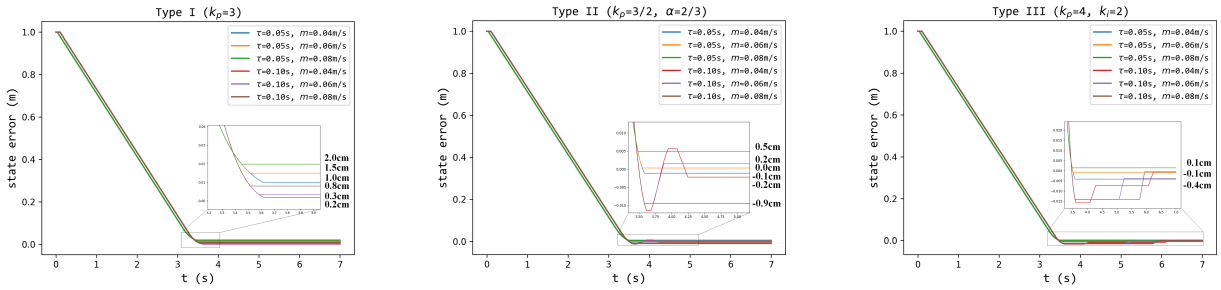
When $k_p^2 - 4k_i \neq 0$, equation (10) has two different roots $\lambda_{1,2} \in \mathbb{C}$. Then we substitute the initial condition $\begin{cases} \epsilon(0) = 0 \\ \dot{\epsilon}(0) = e(0) \end{cases}$ and differentiate ϵ with respect to t, which yields $e = \frac{-\lambda_1}{\lambda_2 - \lambda_1} e(0) \exp\{\lambda_1 t\} + \frac{\lambda_2}{\lambda_2 - \lambda_1} e(0) \exp\{\lambda_2 t\}$, indicating two exponential modes of the system .
By introducing integral term, Type III controller manages to eliminate steady-state error for any dead zone size, exhibiting much stronger robustness to dead zones compared to the Type I controller.

In summary, while the Type I controller is the simplest, it lacks the robustness required for sim-to-real transfer. The Type II controller improves upon Type I in terms of accuracy, with the requirement barely met yet. The Type III controller stands out as the most robust and accurate among the three. In further analysis with transport delay, dead zone and saturation, it provides an accuracy margin of up to 0.5 cm (see Fig. 7 for extensive results).

Therefore, with feedback linearization and Design Function, we manage to build a nonlinearity-robust servo system

[1]For brevity, we abbreviate $\mathbf{e}(t)$ to $\mathbf{e}$ and use $\exp\{\cdot\}$ to denote the exponential function. The notation also holds for x, x_d, etc.

[2]In order to extend the power function to $e(t) < 0$, we only need to let $f(e,t) = -k_p \cdot \text{sign}(e) \cdot |e|^{\alpha}$ and set $\tilde{e} = -e$, which yields symmetric results.



(a) (b) (c)

Fig. 7. Error attenuation curves of the servo system with three DF choices, where the transport delay $\tau \in [0.05, 0.1]$, the velocity dead zone size $m \in [0.04, 0.08]$ and the velocity saturation is 0.3 m/s. Steady-state errors are annotated on each curve. Type I controller (a) is sensitive to dead zone and fails to achieve sub-centimeter accuracy; Type II controller (b) barely meets the requirement; Type III controller (c) meets the requirement with an accuracy margin of 0.5 cm.

that handles inappropriate grasp pose and unmodeled nonlinearities simultaneously, which proves to be effective in the 2024 RSC.

V. Experiments

In this section, we evaluate our visual perception system, servo system, and the entire robotic system within the 2024 RSC simulator and real-world environment. The robot is equipped with Intel® NUC Kit NUC11PAHi7 with an i7-1165G7 (2.8 GHz, 8 cores) processor and 8GB of memory. The RSC simulator runs in Docker containers to match the computing power of real hardware. The real-world test results are based on the final evaluation conducted by the competition committee. Original data and multi-view videos of the evaluation are available at `https://drive.google.com/drive/folders/1yd4S0y6_FNU45j6xjc7f12Y2G_dM1_Cl`.

A. Evaluating Visual Perception System

Evaluation Metrics. Following common practices, we use recall rate, error rate and sample standard deviation as metrics. Recall rate is defined as the ratio of the number of recognized minerals (those rejected by the classifier are excluded) to the total number of minerals of frames. Error rate is the ratio of the number of misclassified contours to the number of the recognized.

Let $\mathbf{p}_i$ and θ_i be the estimated position and orientation angle (the angle between the surface normal and the positive x-axis) of the target mineral in the global coordinate system, with $\bar{\mathbf{p}}$ and $\bar{\theta}$ being their mean values. For the temporal sequence of estimates of length l, we use the sample standard deviation to evaluate its precision:

$$S_{pos} = \sqrt{\frac{1}{l-1}\sum_{i=1}^{l} \|\mathbf{p}_i - \bar{\mathbf{p}}\|_2^2} \tag{11}$$

$$S_{att} = \sqrt{\frac{1}{l-1}\sum_{i=1}^{l} (\theta_i - \bar{\theta})^2} \tag{12}$$

Results. We train our classifier for 1200 epochs with Adam optimizer and set the rejection threshold as 12 to achieve the complete rejection of incorrect contours. Then, we test recall rate, error rate and pose estimation precision within 1 m (see Table I).

TABLE I

Performance of Visual Perception Methods in 2024 RSC Simulated and Real-World Environments

Method	Env	Recall Rate*(%)	Error Rate*(%)	Precision	
				S_{pos}	S_{att}
Vanilla ArUco detector	Sim	48.7	1.1†	1.4 cm	4.8 deg
	Real	49.2	1.1†	1.2 cm	2.1 deg
Ours w/o SMMS	Sim	96.0	0.0†	1.3 cm	1.0 deg
	Real	50.5	0.0†	1.2 cm	2.2 deg
Ours w/ SMMS	Sim	91.6	0.0	0.8 cm	1.0 deg
	Real	70.0	0.0	0.6 cm	1.9 deg

* represents the average of the measurements for mineral markers and the exchange markers.

† represents the results susceptible to detector parameters and the background. (e.g. brown doors in the room where competition holds cause about 5% false recognitions, thereby leading to notable misclassifications)

Compared with vanilla ArUco detector, our perception pipeline increases recall rate of minerals (from 48.7% to 96.0%) and reduces error rate of classification (from 1.1% to 0.0%), both of which provide more valid samples for pose estimation, thereby raising pose estimation precision. However, without SMMS, it suffers great loss of recall rate (from 96.0% to 50.5%), and falls short in dealing with false recognitions caused by visible objects in the background. By applying SMMS, our visual perception system reduces recall rate loss during sim-to-real, and enhances the robustness by rejecting all false recognitions, which yields pose estimation precision within 1 cm and 2 deg.

Additionally, we test the run-time of each component of our pipeline. It shows that the average run-times of our detector, classifier and PnP solver are 6.68, 3.60 and 0.86 ms per frame (11.14 ms per frame in total), exhibiting high efficiency of our visual perception system.

Overall, with proposed SMMS, our visual perception system manages to address measurement noise introduced by motion blur. It mitigates the sharp drop in recall rate and maintains classification accuracy and estimation precision,

demonstrating noise resilience and high consistency during sim-to-real transfer.

B. Evaluating Servo System

We evaluate the accuracy and speed of our servo system using the Type III controller, which is identified as the most robust and accurate in section IV-C. The proportional gain k_p is set to 4, and the integral gain k_i is set to 2, with an integral separation threshold[3] of 0.1 m.

Performance metrics. We use final error and average aligning time as performance metrics. For each servo adjustment, we first define the termination condition as the position error remaining within the tolerable range for 2 seconds, with horizontal and vertical tolerances set to 0.015m. Then, we measure aligning time (the interval from the beginning to termination) and final error (the position or attitude error when terminated, and it is marked as '-' if not aligned by the controller).[4]

Results. Table II shows the average servo error and aligning time. The open-loop controller fails overcoming actual disturbance in dynamics, while the PID controller falls short in aligning coupled position and attitude simultaneously, both of which fail in real grasping. With feedback linearization and DF, our servo system with Type III controller achieves sub-centimeter accuracy with tolerable aligning time of about 4 seconds. Overall, it manages to overcome sensing discrepancies with comparable performance between simulation and reality.

TABLE II
PERFORMANCE OF SERVO CONTROL METHODS IN 2024 RSC SIMULATED AND REAL-WORLD ENVIRONMENTS

Method	Env	Final Error			Aligning Time (s)	Result
		x (cm)	y (cm)	θ (deg)		
Open-loop*	Sim	1.24	0.13	-	2.16	✓
	Real	7.11	4.07	-	Timeout	×
PID*	Sim	0.27	0.22	-	2.18	✓
	Real	0.35	0.86	-	Timeout	×
Ours (Type III)	Sim	0.48	0.40	0.61	4.06	✓
	Real	0.81	0.63	2.18	4.12	✓

* represents unable to align attitude while aligning position.

C. Full System Evaluation in 2024 RSC

We evaluate the grasping and placing performance of the entire robotic system in both simulation and reality.

Results. Table III shows the ratio of successful grasps to total attempts during the grasping and placing processes. During the final evaluation of the 2024 RSC, our system achieved a 100% success rate in both grasping and stacking tasks, being the only team to successfully complete both two-layer and three-layer stacking.

Postmortem. We observed approximately a 1/6 failure rate in placing tasks in simulation. Further analysis reveals that these failures are due to the minerals falling after being stacked. This issue arises because the simulator cannot handle collisions effectively, leading to mineral jittering and dropping.

TABLE III
PICK-AND-PLACE SUCCESS RATE IN 2024 RSC SIMULATED AND REAL-WORLD ENVIRONMENTS

Environment	Grasp Success	Stack Success
Simulator	30/30	22/30
Reality	6/6	6/6

VI. CONCLUSION AND DISCUSSION

Since the introduction of simulators, researchers have continuously pursued advancements in sim-to-real transfer. Pick-and-place tasks, which feature rich interactions with scene objects, further complicate this challenge. Sim-to-real transfer remains an unresolved issue in the field of embodied intelligence.

In this context, our robotic system demonstrates that it is both possible and feasible to achieve consistent performance between simulation and reality in long-horizon pick-and-place tasks: by applying **Sequential Motion-blur Mitigation Strategy (SMMS)**, our visual perception system manages to address motion blur and provide consistently precise pose estimates; by applying feedback linearization and introducing **Design Function (DF)**, our servo system manages to address inappropriate grasp pose as well as unmodeled nonlinearities, and maintain sub-centimeter level accuracy of servo control. Additionally, we provide extensive analysis and comparison of three intuitive DF choices and conclude that the Type III demonstrates required robustness to nonlinearities. Benefiting from these, our robotic system successfully completed long-horizon pick-and-place tasks both in simulator and in real-world, securing first place in the 2024 Robotic Sim2Real Challenge [30].

[3]The integral term only works within given threshold.

[4]As the robot is near the target and almost stationary at the end of the alignment, measurement error introduced by the visual perception system is negligible.

REFERENCES

[1] Ananye Agarwal et al. "Dexterous functional grasping". In: *7th Annual Conference on Robot Learning*. 2023.

[2] Karol Arndt et al. "Meta reinforcement learning for sim-to-real domain adaptation". In: *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2020, pp. 2725–2731.

[3] Hansenclever F Bassani et al. "Learning to play soccer by reinforcement and applying sim-to-real to compete in the real world". In: *arXiv preprint arXiv:2003.11102* (2020).

[4] Yuqing Du et al. "Auto-tuned sim-to-real transfer". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 1290–1296.

[5] Karl Dietrich von Ellenrieder. "Feedback Linearization". In: *Control of Marine Vehicles*. Cham: Springer International Publishing, 2021, pp. 315–363. ISBN: 978-3-030-75021-3. DOI: 10.1007/978-3-030-75021-3_8. URL: https://doi.org/10.1007/978-3-030-75021-3_8.

[6] Sergio Garrido-Jurado et al. "Automatic generation and detection of highly reliable fiducial markers under occlusion". In: *Pattern Recognition* 47.6 (2014), pp. 2280–2292.

[7] Yinlin Hu et al. "Segmentation-driven 6d object pose estimation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 3385–3394.

[8] Shariq Iqbal et al. "Toward sim-to-real directional semantic grasping". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 7247–7253.

[9] Stephen James, Marc Freese, and Andrew J. Davison. "PyRep: Bringing V-REP to Deep Robot Learning". In: *arXiv preprint arXiv:1906.11176* (2019).

[10] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. *Ultralytics YOLO*. Version 8.0.0. Jan. 2023. URL: https://github.com/ultralytics/ultralytics.

[11] Oguz Kedilioglu et al. "ArUcoE: enhanced ArUco marker". In: *2021 21st International Conference on Control, Automation and Systems (ICCAS)*. IEEE. 2021, pp. 878–881.

[12] Donghyeon Kim et al. "Torque-based deep reinforcement learning for task-and-robot agnostic learning on bipedal robots using sim-to-real transfer". In: *IEEE Robotics and Automation Letters* (2023).

[13] Eric Kolve et al. "Ai2-thor: An interactive 3d environment for visual ai". In: *arXiv preprint arXiv:1712.05474* (2017).

[14] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[15] Edwin Olson. "AprilTag: A robust and flexible visual fiducial system". In: *2011 IEEE international conference on robotics and automation*. IEEE. 2011, pp. 3400–3407.

[16] Ole-Magnus Pedersen. "Sim-to-real transfer of robotic gripper pose estimation-using deep reinforcement learning, generative adversarial networks, and visual servoing". MA thesis. NTNU, 2019.

[17] Xue Bin Peng et al. "Sim-to-real transfer of robotic control with dynamics randomization". In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 3803–3810.

[18] Shaoqing Ren et al. "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *IEEE transactions on pattern analysis and machine intelligence* 39.6 (2016), pp. 1137–1149.

[19] Manolis Savva et al. "Habitat: A platform for embodied ai research". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 9339–9347.

[20] Manolis Savva et al. "MINOS: Multimodal indoor simulator for navigation in complex environments". In: *arXiv preprint arXiv:1712.03931* (2017).

[21] Josh Tobin et al. "Domain randomization for transferring deep neural networks from simulation to the real world". In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2017, pp. 23–30.

[22] Jeroen Van Baar et al. "Sim-to-real transfer learning using robustified controllers in robotic tasks involving complex dynamics". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 6001–6007.

[23] Jiayu Wang et al. "Learning semantic keypoint representations for door opening manipulation". In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 6980–6987.

[24] John Wang and Edwin Olson. "AprilTag 2: Efficient and robust fiducial detection". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 4193–4198.

[25] Yi Wu et al. "Building generalizable agents with a realistic and rich 3d environment". In: *arXiv preprint arXiv:1801.02209* (2018).

[26] Fanbo Xiang et al. "Sapien: A simulated part-based interactive environment". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11097–11107.

[27] Pengwei Xie et al. "Part-guided 3D RL for Sim2Real articulated object manipulation". In: *IEEE Robotics and Automation Letters* (2023).

[28] Xiang Zhang, Masayoshi Tomizuka, and Hui Li. "Bridging the Sim-to-Real Gap with Dynamic Compliance Tuning for Industrial Insertion". In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 4356–4363.

[29] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. "Sim-to-real transfer in deep reinforcement learning for robotics: a survey". In: *2020 IEEE symposium series on computational intelligence (SSCI)*. IEEE. 2020, pp. 737–744.

[30] Guyue Zhou et al. *The 3rd Robotic Sim2Real Challenges*. http://www.sim2real.net/. [Accessed 13-09-2024]. 2024.