

Article

Fusing Phase Map Servoing and MPC for High-Precision Robotic Tracking of Dynamic Objects

Qinghui Zhang ¹, Tianhao Han ², Lei Lu ^{3,*}, Wei Pan ^{4,*}  and Ge Gao ⁵¹ College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China; zhangqinghui1974@163.com² College of Electrical Engineering, Henan University of Technology, Zhengzhou 450001, China; tianhaohan@stu.haut.edu.cn³ Institute for Complexity Science, Henan University of Technology, Zhengzhou 450001, China⁴ Department of R&D, OPT Machine Vision Tech Co., Ltd., Dongguan 523860, China⁵ Department of R&D, Mech-Mind Robotics Technologies Ltd., Beijing 100085, China; gao.ge@mech-mind.net

* Correspondence: lulei@haut.edu.cn (L.L.); vpan@foxmail.com (W.P.)

Abstract

This paper presents a unified framework for high-precision dynamic target tracking that combines phase-map-based visual servoing with Model Predictive Control (MPC). Phase maps obtained from fringe projection provide dense, subpixel geometric feedback, enabling accurate end-effector velocity computation; however, their high dimensionality leads to substantial computational overhead that hinders real-time control. To overcome this limitation, we introduce a phase-map-specific dimensionality reduction strategy that constructs a low-dimensional control subspace through gradient-guided sparsification and PCA embedding while preserving the controllability of the original interaction model. An adaptive prediction horizon is further developed to regulate MPC complexity according to the rate of phase variation, enabling real-time deployment without compromising tracking accuracy. In addition, an Extended Kalman Filter (EKF) is integrated into the control loop to compensate for system delays and improve trajectory prediction in dynamic scenarios. Experimental results on multi-axis robotic manipulation demonstrate that the proposed approach achieves superior tracking accuracy and computational efficiency compared with conventional visual servoing methods, validating the feasibility of phase-map-driven predictive control for high-speed dynamic target tracking.

Keywords: visual servoing; phase difference; servo control rate; Model Predictive Control (MPC); dimensionality-reduced modeling; dynamic object



Academic Editor: Zhuming Bi

Received: 15 December 2025

Revised: 13 January 2026

Accepted: 23 January 2026

Published: 28 January 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Real-time tracking of dynamic objects is becoming increasingly important in modern industrial automation. Traditional robotic workcells typically rely on stop-and-go strategies in which the robot moves to a fixed pose, acquires an image, and then executes a predefined task. Although robust, such pipelines fundamentally limit throughput and cannot accommodate next-generation intelligent manufacturing systems, where objects move continuously on conveyor belts or mobile platforms. Visual servoing (VS) provides a promising direction by using continuous visual feedback to close the control loop, enabling robots to react to real-time changes in the scene. Classical VS approaches—such as image-based visual servoing (IBVS), position-based visual servoing (PBVS), and hybrid formulations—have been extensively studied and successfully deployed in manipulation,

inspection, and assembly [1,2]. The development of high-speed cameras and modern computation has further enabled increasingly complex real-time strategies [3,4]. Early studies demonstrated the feasibility of visual servoing for dynamic target tracking using image feedback, particularly with pan-tilt camera systems [5,6]. These methods were later extended to mobile robots with nonholonomic constraints, highlighting the challenges of coupling perception and motion in dynamic environments [7]. Motion-based visual servoing further strengthened the relationship between image motion and control actions, improving robustness in tracking tasks [8]. More recent work has applied visual servoing to highly dynamic platforms such as UAVs and incorporated robust predictive control strategies to explicitly handle system constraints and visibility limitations [9]. These studies collectively reveal an increasing demand for more robust visual representations and control frameworks in dynamic visual servoing applications.

Building upon these developments, the theoretical foundations of visual servoing have been well established through image-based formulations and interaction matrix analysis, which guarantee closed-loop stability when appropriate visual features are employed [10–12]. Extensions toward optimal and predictive control further improved tracking performance by explicitly accounting for future motion and system constraints, particularly in constrained image-based visual servoing frameworks [13]. Meanwhile, phase-based and fringe-based visual representations have demonstrated the ability to capture dense motion and surface information with sub-pixel sensitivity, offering a robust alternative to sparse feature-based methods, especially under low-texture or high-dynamic conditions [14–16]. These studies collectively motivate the integration of phase-based visual features with predictive visual servoing to achieve reliable and precise dynamic target tracking.

Despite their success, conventional visual features (points, edges, contours) struggle on textureless or specular surfaces that frequently occur in industrial environments. Phase maps obtained from fringe projection profilometry have recently emerged as a compelling alternative because they offer dense, subpixel-resolution geometry without requiring 3D reconstruction. This work integrates classical visual servoing theory with phase-based visual representations to construct a closed-loop control framework for dynamic target tracking. Fundamental visual servoing methods, such as image Jacobian modeling and error mapping, provide theoretical guarantees for the stability and convergence of the control law, enabling arbitrary visual features to be systematically incorporated into the control loop [1]. Meanwhile, phase-based approaches analyze local phase information across multi-scale pyramids and narrow-band filters, allowing sub-pixel motion to be captured and amplified, and yielding denser and more robust displacement representations than traditional optical flow or keypoint-based tracking methods [17]. Furthermore, phase-based registration techniques, such as phase-only correlation (POC), together with phase filtering strategies designed for complex illumination and weak-texture scenarios, offer practical engineering solutions for stable error quantification and control-region masking in industrial and dynamic environments [18]. Overall, introducing phase-map features into visual servoing not only enables more stable dense motion estimation under weak-texture, low-contrast, and high-frequency vibration conditions, but also provides inherently low-latency and high-sensitivity multi-scale representations that are particularly well suited for closed-loop tracking of fast or periodic moving targets, thereby significantly improving the success rate and accuracy of dynamic object capture and tracking tasks.

Prior studies show that phase-map interaction models enable high-precision servo control on curved and textureless surfaces [19,20]. However, existing phase-map-based control laws have been primarily applied to static or slowly varying scenes. Although recent progress has demonstrated the ability of fringe projection systems to reconstruct deforming or moving objects [21,22], the direct use of phase maps for closed-loop control in dynamic scenarios remains largely unexplored.

Dynamic target tracking introduces two additional challenges. First, system latency and image processing delays cause deviations between the acquired phase map and the actual object state. These delays accumulate at higher target velocities, degrading control accuracy or even inducing instability. Second, achieving high-frequency predictive control using dense phase measurements is computationally intensive. Model Predictive Control (MPC) is theoretically well suited to dynamic environments due to its ability to incorporate constraints, explicitly model future trajectories, and handle multi-objective optimization [23,24]. However, the standard MPC formulation becomes prohibitively expensive when applied directly to thousands of visual features extracted from phase maps, forcing practitioners to reduce the prediction horizon or update rate [25,26]. Although various accelerated solvers such as OSQP, qpOASES, and CVXGEN have been proposed [27], the computational bottleneck persists when the dimensionality of the visual feature space is large.

To address latency, prediction, and computational limitations simultaneously, this paper proposes a unified framework that integrates phase-map-based visual servoing with MPC and Extended Kalman Filtering (EKF). The key insight is that phase maps, although high-dimensional, possess strong geometric structure. By exploiting phase gradients and intrinsic spatial correlations, we develop a principled dimensionality reduction strategy that preserves the controllability of the original visual servo law. The resulting reduced-order model significantly lowers the computational cost of MPC while maintaining accuracy. Furthermore, an adaptive prediction horizon is introduced to balance responsiveness and runtime, and EKF prediction compensates for system latency and improves trajectory estimation.

The primary contributions of this work are as follows:

- We introduce the first unified framework that applies phase-map measurements directly to dynamic visual servoing with predictive control.
- We propose a phase-map-specific dimensionality reduction technique that combines gradient-induced sparsification and PCA-based low-dimensional embedding while preserving visual controllability.
- We design an adaptive horizon MPC formulation that adjusts prediction depth based on feature dynamics, enabling real-time execution.
- We integrate EKF-based motion prediction into the visual servo loop, reducing tracking errors caused by latency.
- We validate the proposed method through both simulation and physical experiments on a dynamic object-grasping task, demonstrating improved accuracy and computational efficiency over conventional visual servoing.

Related Work

Visual servoing has long served as a fundamental paradigm for controlling robotic manipulators using visual feedback. Early works established the foundations of image-based visual servoing (IBVS) and position-based visual servoing (PBVS), offering complementary strengths in stability, robustness, and modeling requirements [1,2]. IBVS avoids explicit 3D reconstruction by directly regulating features in the image plane, whereas PBVS leverages 3D models to improve convergence properties. Hybrid strategies combine elements of both frameworks to overcome their respective limitations. Recent advancements in

sensing and computation have enabled predictive, learning-based, and multi-rate extensions of classical visual servoing [28–31], expanding its applicability to high-speed and uncertain environments.

Recent advances in robotic visual servoing have extended classical frameworks to handle dynamic and complex environments by integrating predictive estimation and control strategies. Adaptive extended Kalman filter (EKF)-based approaches have been shown to enable stable tracking of non-cooperative or fast-moving targets, enhancing the robustness of visual feedback under sensor and motion uncertainties [32]. Similarly, predictive control techniques, including model predictive control (MPC) for pan-tilt cameras and manipulators, have demonstrated improved tracking accuracy by explicitly considering future motion trajectories and control constraints [33]. Moreover, recent developments combining high-rate 6D object pose estimation and closed-loop MPC facilitate precise real-time guidance for dynamic target manipulation [34]. Despite these advances, many methods rely on sparse feature points or traditional image-based representations, which may be insufficient in low-texture, high-speed, or weak-contrast scenarios. By incorporating phase-based visual representations into the servoing loop, subtle motions can be captured at sub-pixel precision, enabling dense, robust, and low-latency feedback suitable for dynamic object tracking. Therefore, integrating phase-based visual cues with predictive control strategies is a necessary step toward improving the precision and reliability of closed-loop robotic manipulation in dynamic environments [35–38].

Phase-map-based visual servoing represents a more recent direction aimed at achieving precise geometry-aware control without requiring feature extraction. Using fringe projection profilometry, phase maps encode high-resolution surface information that can be directly mapped to end-effector motion through analytically derived interaction matrices. Xu et al. [19] demonstrated phase-based servoing for curved surfaces, and Li et al. [20] extended this approach to cylindrical welding. Despite these successes, prior work has focused on static or quasi-static scenes and thus does not address challenges posed by dynamic object motion. Although phase-based reconstruction for dynamic objects has progressed significantly [21,22], these developments have not yet translated into real-time control frameworks.

Model Predictive Control (MPC) has gained substantial traction in robotics due to its ability to incorporate constraints, predict future states, and optimize nonconvex behaviors [23,24]. MPC has been used for manipulator trajectory tracking, mobile robot navigation, multi-contact motion planning, and whole-body control [4,39,40]. However, the application of MPC to vision-based control is limited by the dimensionality of visual features. Standard MPC requires repeated construction of large prediction matrices and solves convex optimization problems at high frequency; consequently, computational load increases sharply as feature dimensionality grows [25]. While efficient solvers such as qpOASES, HPIPM, OSQP, and CVXGEN have reduced per-iteration cost [23,27], computational scalability remains a major obstacle when integrating dense visual data.

To mitigate this issue, several works have explored dimensionality reduction or structure exploitation. PCA-based model reduction has been applied to robot dynamics and whole-body MPC to preserve dominant modes while reducing optimization size [25]. Sparse QP formulations that exploit block or band structures have also been shown to accelerate optimization. However, these methods have not been adapted to the unique characteristics of phase-map interaction models, where phase gradients naturally reveal informative and uninformative regions of the surface. Consequently, existing reduction strategies do not directly apply to phase-based visual servoing. Although this work focuses on robotic visual servoing, the proposed PCA-based dimensionality reduction strategy is not limited to robotic applications. In general, PCA provides an effective way to ex-

tract dominant structures from high-dimensional, spatially correlated sensor data while preserving essential system information. Such characteristics are also desirable in other domains, for example, automated fault detection and diagnosis (AFDD) in building energy systems [41,42].

To the best of our knowledge, this work presents the first cohesive framework that integrates phase-map visual servoing, MPC, EKF prediction, dimensionality reduction, and adaptive horizon selection for dynamic object tracking. By exploiting the intrinsic structure of phase maps, the proposed method reduces computational complexity while preserving the geometric properties essential for visual servo control.

2. Theoretical Foundations

This section presents the theoretical background required for constructing the proposed control framework. We first review the phase-map formation model obtained from fringe projection profilometry, and then derive the interaction matrix that relates phase variations to camera motion. All variables appearing in the mathematical expressions are explicitly defined to ensure clarity, reproducibility, and consistency. The experimental setup is illustrated in Figure 1, the overall framework is shown in Figure 2, and the schematic of phase-map acquisition is presented in Figure 3. The parameters appearing in these figures will be described in detail in the subsequent theoretical derivations.

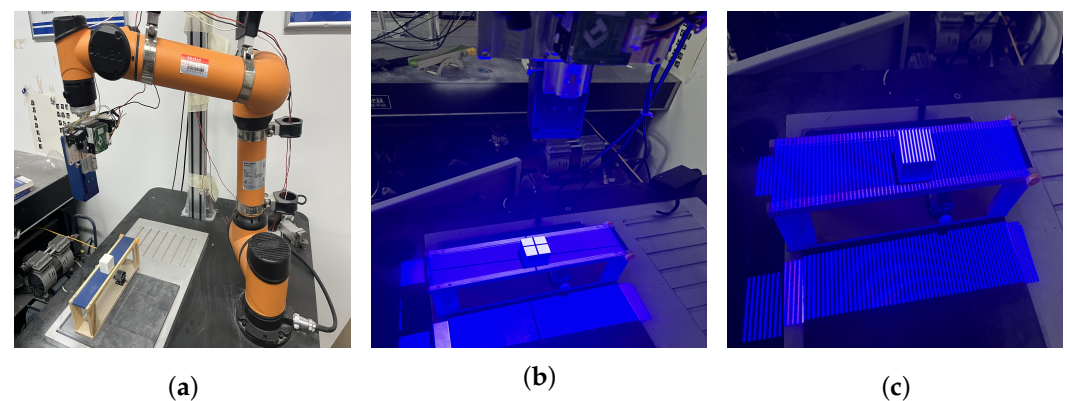


Figure 1. Dynamic Object Tracking Platform. (a) Overall experimental environment. (b) Camera projector and transmission platform. (c) Projection stripe display.

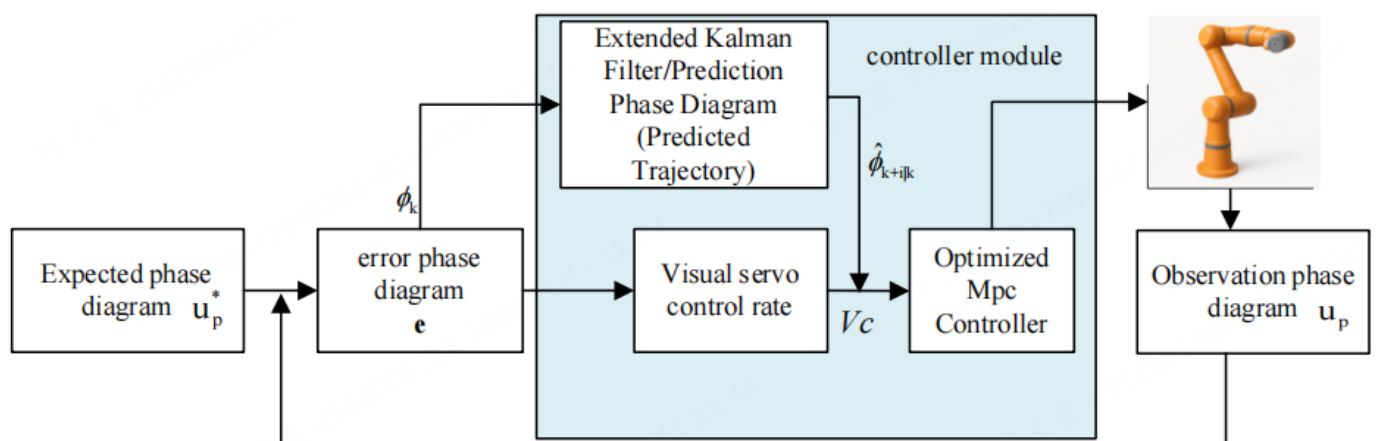


Figure 2. Control framework combining visual servo and MPC.

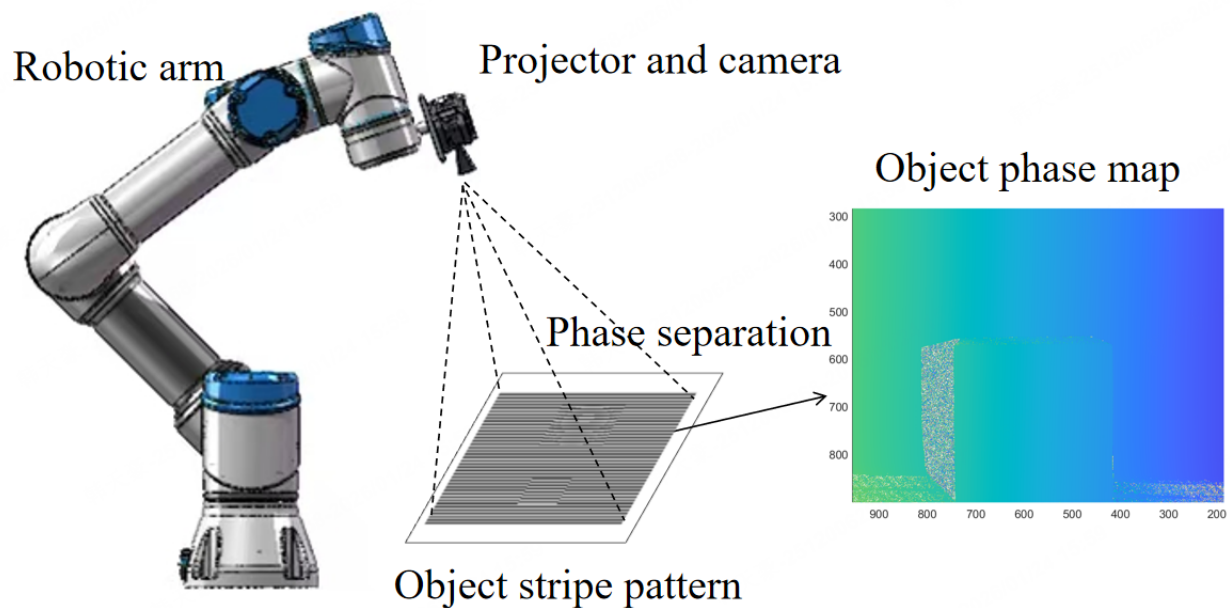


Figure 3. Phase map acquisition diagram.

2.1. Principle of Phase-Mapping Imaging

Fringe Projection Profilometry projects sinusoidal patterns onto the surface of an object. The overall process is illustrated in Figure 3. The observed intensity at pixel coordinates (u_c, v_c) is typically expressed as

$$I(u_c, v_c) = I_0 + I_m \cos(\omega x + \phi(u_c, v_c)), \quad (1)$$

where $\phi(u_c, v_c)$ denotes the wrapped phase. After phase demodulation and unwrapping, a dense phase map

$$u_p = \phi(u_c, v_c)$$

is obtained, providing subpixel-level geometric information about the surface.

In this work, the phase value u_p serves directly as the visual feature for control, avoiding feature extraction or reconstruction steps. Let the vectorized phase map be denoted by

$$\mathbf{u}_p = [u_p^{(1)}, u_p^{(2)}, \dots, u_p^{(m)}]^\top \in \mathbb{R}^m,$$

where m is the number of selected phase pixels.

2.2. Phase-Map-Based Visual Servo Control Rate

The temporal evolution of the phase map is related to the camera (or end-effector) spatial velocity $\mathbf{V}_c \in \mathbb{R}^6$, defined as

$$\mathbf{V}_c = [v_x \ v_y \ v_z \ \omega_x \ \omega_y \ \omega_z]^\top.$$

The fundamental interaction equation is given by

$$\dot{\mathbf{u}}_p = \mathbf{L}_{u_p} \mathbf{V}_c, \quad (2)$$

where $\dot{\mathbf{u}}_p$ is the measured time derivative of the phase map, and $\mathbf{L}_{u_p} \in \mathbb{R}^{m \times 6}$ is the interaction matrix associated with the phase feature.

To avoid ambiguity between intrinsic temporal variation and motion-induced variation, we separate the true temporal derivative of phase

$$\dot{u}_p^{\text{true}} = \frac{\partial u_p}{\partial t}$$

from the apparent phase change caused by camera or object motion. Using the chain rule, we obtain

$$\dot{u}_p^{\text{true}} = \dot{u}_p + G_{u_c} \dot{u}_c + G_{v_c} \dot{v}_c, \quad (3)$$

where

$$G_{u_c} = \frac{\partial u_p}{\partial u_c}, \quad G_{v_c} = \frac{\partial u_p}{\partial v_c}$$

are the spatial gradients of the phase map.

Equation (3) is consistent with FPP geometry: motion in the image plane induces additional apparent phase variation proportional to the local phase gradients.

2.2.1. Interaction Matrix Decomposition

Following [19,20], the per-pixel interaction model is formulated using a unified notation. The common geometric terms used throughout this section are defined as

$$B = \frac{1}{Z}, \quad J = r_{31}u_c + r_{32}v_c + r_{33}, \quad K_i = r_{1i} - u_p r_{3i}, \quad i = 1, 2, 3, \quad (4)$$

where Z denotes the depth of the scene point from the camera, r_{ij} are the elements of the rotation matrix $\mathbf{R}_{pc}$ from the projector frame to the camera frame, and t_3 is the z-component of the translation vector $\mathbf{t}_{pc}$.

The per-pixel interaction matrix $\mathbf{L}_{u_p}$ is decomposed into three components as

$$\mathbf{L}_{u_p} = \mathbf{L}_{du} - G_{u_c} \mathbf{L}_{u_c} - G_{v_c} \mathbf{L}_{v_c}, \quad (5)$$

where $\mathbf{L}_{du}$ describes the intrinsic phase response to 3D motion, $\mathbf{L}_{u_c}$ and $\mathbf{L}_{v_c}$ correspond to image-plane motions along u_c and v_c , and G_{u_c} and G_{v_c} denote the phase gradients.

The closed-form expressions of the six interaction terms are given by

$$L_{u_p}^{v_x} = -\frac{K_1 B}{J + t_3 B} + G_{u_c} B, \quad (6)$$

$$L_{u_p}^{v_y} = -\frac{K_2 B}{J + t_3 B} + G_{v_c} B, \quad (7)$$

$$L_{u_p}^{v_z} = -\frac{K_3 B}{J + t_3 B} - G_{u_c} u_c B - G_{v_c} v_c B, \quad (8)$$

$$L_{u_p}^{\omega_x} = \frac{K_2 - K_3 v_c}{J + t_3 B} - G_{u_c} u_c v_c - G_{v_c} (1 + v_c^2), \quad (9)$$

$$L_{u_p}^{\omega_y} = \frac{K_3 u_c - K_1}{J + t_3 B} + G_{u_c} (1 + u_c^2) + G_{v_c} u_c v_c, \quad (10)$$

$$L_{u_p}^{\omega_z} = \frac{K_1 v_c - K_2 u_c}{J + t_3 B} - G_{u_c} v_c + G_{v_c} u_c. \quad (11)$$

Stacking all pixel-wise interaction terms yields the full interaction matrix

$$\mathbf{L}_{u_p} = \begin{bmatrix} \mathbf{L}_{u_p}^{(1)} \\ \mathbf{L}_{u_p}^{(2)} \\ \vdots \\ \mathbf{L}_{u_p}^{(m)} \end{bmatrix} \in \mathbb{R}^{m \times 6}. \quad (12)$$

2.2.2. Control Law

For implementation clarity, the visual-servoing reference velocity can be written in explicit form as

$$\mathbf{V}_c = -\lambda \mathbf{L}_{u_p}^+(\mathbf{u}_p - \mathbf{u}_p^*) \quad (13)$$

where $\lambda > 0$ is the control gain and $\mathbf{L}_{u_p}^+$ denotes the Moore–Penrose pseudoinverse. In this work, $\mathbf{V}_c \in \mathbb{R}^6$ represents a Cartesian task-space velocity (linear and angular components), which is directly sent to the robot controller.

Based on the control law given in (13), experiments were conducted to evaluate the visual servoing performance. The results are shown in Figure 4, which illustrates the variation of the phase map under the proposed control law.

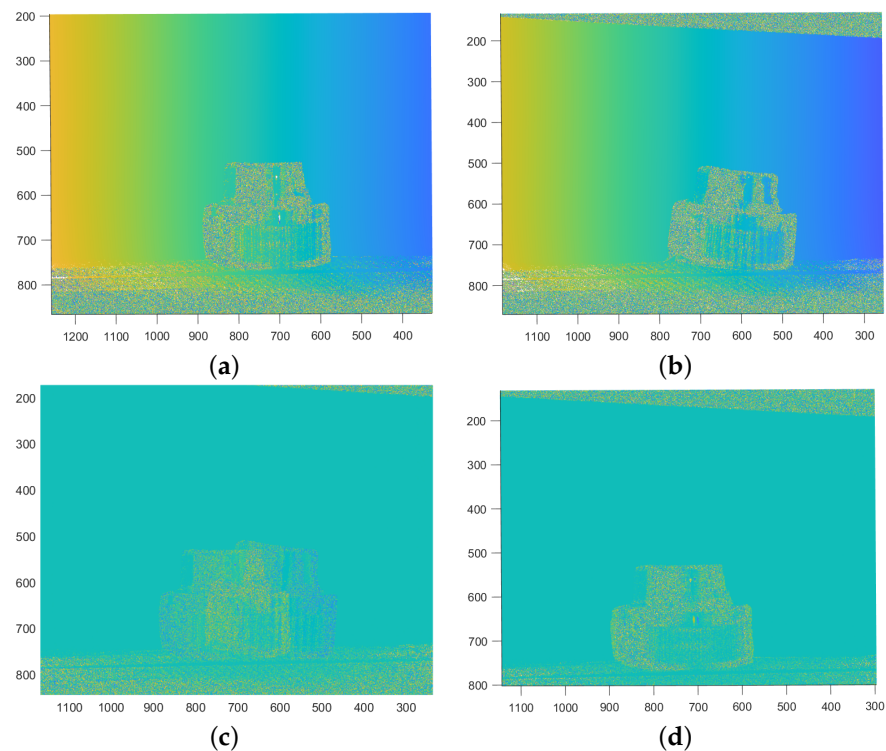


Figure 4. Phase map changes and tracking curves for the visual error index decline process. (a) Expected phase map. (b) Actual phase map. (c) Initial phase map error (horizontally mirrored). (d) Final phase map error (horizontally mirrored).

2.3. Modeling of Robotic Arms Under Model Predictive Control

Model Predictive Control (MPC) regulates system behavior by repeatedly solving a finite-horizon optimization problem. In the context of phase-map visual servoing, the state variable can be naturally defined as the phase error

$$\phi_k = \mathbf{u}_p(k) - \mathbf{u}_p^*,$$

where $\mathbf{u}_p^*$ denotes the desired phase map. The control input is the camera (or end-effector) velocity

$$\mathbf{V}_c(k) = \dot{\mathbf{x}}_k \in \mathbb{R}^6.$$

2.3.1. Continuous-Time Model

Using the interaction relation (2), the continuous-time evolution of the phase error is expressed as

$$\dot{\phi}(t) = \mathbf{L}_{u_p}(t) \mathbf{V}_c(t), \quad (14)$$

where $\mathbf{L}_{u_p}(t)$ is updated online according to Section 2.2.

Since the sampling period Δt of fringe acquisition and controller update is small (typically 1–5 ms), the interaction matrix varies slowly within each sampling interval. Thus, standard discretization via Forward Euler is appropriate and introduces negligible temporal approximation error.

2.3.2. Discrete-Time Model

Applying Forward Euler to (14) yields

$$\boldsymbol{\phi}_{k+1} = \boldsymbol{\phi}_k + \mathbf{L}_{u_p}(k) \mathbf{V}_c(k) \Delta t. \quad (15)$$

Let us denote the control input as

$$\mathbf{u}_k = \mathbf{V}_c(k),$$

and define the time-varying system matrices

$$\mathbf{A}_k = \mathbf{I}, \quad \mathbf{B}_k = \mathbf{L}_{u_p}(k) \Delta t.$$

Then (15) becomes

$$\boldsymbol{\phi}_{k+1} = \mathbf{A}_k \boldsymbol{\phi}_k + \mathbf{B}_k \mathbf{u}_k. \quad (16)$$

2.3.3. Prediction Model

Over a prediction horizon N , the stacked future states satisfy

$$\begin{aligned} \boldsymbol{\Phi} = & \underbrace{\begin{bmatrix} \mathbf{A}_k \\ \mathbf{A}_{k+1} \mathbf{A}_k \\ \vdots \\ \mathbf{A}_{k+N-1} \cdots \mathbf{A}_k \end{bmatrix}}_{\mathbf{S}_\phi} \boldsymbol{\phi}_k \\ & + \underbrace{\begin{pmatrix} \mathbf{B}_k & 0 & \cdots & 0 \\ \mathbf{A}_{k+1} \mathbf{B}_k & \mathbf{B}_{k+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{k+N-1} \cdots \mathbf{A}_{k+1} \mathbf{B}_k & \mathbf{A}_{k+N-1} \cdots \mathbf{A}_{k+2} \mathbf{B}_{k+1} & \cdots & \mathbf{B}_{k+N-1} \end{pmatrix}}_{\mathbf{T}} \mathbf{U}. \end{aligned} \quad (17)$$

where

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_{k+1}^\top & \boldsymbol{\phi}_{k+2}^\top & \cdots & \boldsymbol{\phi}_{k+N}^\top \end{bmatrix}^\top,$$

and the control sequence is

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_k^\top & \mathbf{u}_{k+1}^\top & \cdots & \mathbf{u}_{k+N-1}^\top \end{bmatrix}^\top \in \mathbb{R}^{6N}.$$

Because $\mathbf{A}_k = \mathbf{I}$, the prediction model simplifies to

$$\boldsymbol{\Phi} = \mathbf{1}_N \otimes \boldsymbol{\phi}_k + \mathbf{T} \mathbf{U}, \quad (18)$$

where $\mathbf{T}$ becomes a lower block-triangular matrix with repeated $\mathbf{B}_k$ terms.

This structure is advantageous as it preserves the sparsity pattern introduced by the reduced and sparsified interaction matrix.

2.3.4. Cost Function

The MPC objective penalizes both the predicted phase error and the control effort:

$$J(\mathbf{U}) = \sum_{i=1}^N \left\| \boldsymbol{\phi}_{k+i} - \boldsymbol{\phi}_{k+i}^{\text{ref}} \right\|_{\mathbf{Q}}^2 + \sum_{i=0}^{N-1} \left\| \mathbf{u}_{k+i} \right\|_{\mathbf{R}}^2, \quad (19)$$

where:

- $\mathbf{Q} \succeq 0$ is the phase-error weighting,
- $\mathbf{R} \succ 0$ is the control penalty,
- $\boldsymbol{\phi}_{k+i}^{\text{ref}}$ is the reference phase trajectory (from EKF prediction).

Substituting (18) into (19) yields the standard QP form:

$$\min_{\mathbf{U}} \frac{1}{2} \mathbf{U}^{\top} \mathbf{H} \mathbf{U} + \mathbf{g}^{\top} \mathbf{U}, \quad (20)$$

where the Hessian $\mathbf{H}$ and gradient $\mathbf{g}$ are constructed from $\mathbf{T}$, $\mathbf{Q}$ and $\mathbf{R}$.

2.3.5. Constraints

Velocity limits and acceleration bounds are included as follows:

$$\mathbf{u}_{\min} \leq \mathbf{u}_{k+i} \leq \mathbf{u}_{\max}, \quad (21)$$

$$\left\| \mathbf{u}_{k+i} - \mathbf{u}_{k+i-1} \right\|_{\infty} \leq \alpha_{\max}, \quad i = 1, \dots, N-1, \quad (22)$$

where:

- $\mathbf{u}_{\min}, \mathbf{u}_{\max}$: joint Cartesian velocity bounds,
- $\alpha_{\max}$: maximum admissible velocity increment (acceleration proxy).

These inequalities can be compactly expressed as

$$\mathbf{A} \mathbf{U} \leq \mathbf{b}, \quad (23)$$

where $\mathbf{A}$ is block-sparse due to the banded structure of the acceleration constraints.

3. Integrated Control Framework

3.1. Complete MPC Formulation

Combining (20) and (23), the MPC problem is written as:

$$\begin{aligned} \mathbf{U}^* = \arg \min_{\mathbf{U}} \quad & \frac{1}{2} \mathbf{U}^{\top} \mathbf{H} \mathbf{U} + \mathbf{g}^{\top} \mathbf{U} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{U} \leq \mathbf{b}, \end{aligned} \quad (24)$$

and only the first control input $\mathbf{u}_k^*$ is applied (receding horizon control).

The block-triangular structure of $\mathbf{T}$ and the sparsity of $\mathbf{A}$ allow efficient solving using structure-exploiting QP solvers such as qpOASES, which support warm-starting between consecutive MPC iterations, significantly reducing computation time.

3.2. Phase-Map-Specific Dimensionality Reduction and Adaptive Horizon MPC

The dense phase map typically contains thousands of pixels. Directly constructing the full interaction matrix $\mathbf{L}_{u_p} \in \mathbb{R}^{m \times 6}$ leads to excessive computational load in the MPC module, as both the prediction matrix $\mathbf{T}$ and the Hessian $\mathbf{H}$ scale linearly with m . In this section, we introduce two complementary techniques: (i) phase-map-specific dimension-

ality reduction, and (ii) adaptive horizon selection driven by feature dynamics. Both are designed to preserve control accuracy while improving computational efficiency.

3.2.1. Phase-Map Gradient-Induced Sparsification

Unlike generic image features, a phase map encodes smoothly varying geometric information. Let

$$G_{u_c} = \frac{\partial u_p}{\partial u_c}, \quad G_{v_c} = \frac{\partial u_p}{\partial v_c}$$

denote the per-pixel phase gradients. Based on (5), the dominant terms of each row $\mathbf{L}_{u_p}^{(i)}$ of the interaction matrix originate from the local surface orientation encoded by the gradients.

Pixels with negligible gradients,

$$\sqrt{G_{u_c}^2 + G_{v_c}^2} < \epsilon,$$

carry little information about the spatial motion, and their contribution to the visual observability is minimal. Therefore, we define a sparsification mask

$$\mathcal{I} = \left\{ i \mid \sqrt{G_{u_c}^{(i)2} + G_{v_c}^{(i)2}} \geq \epsilon \right\}, \quad (25)$$

and retain only the rows of $\mathbf{L}_{u_p}$ indexed by $\mathcal{I}$. The resulting reduced interaction matrix is

$$\tilde{\mathbf{L}}_{u_p} = \mathbf{L}_{u_p}(\mathcal{I},) \in \mathbb{R}^{m_r \times 6}, \quad m_r = |\mathcal{I}| \ll m.$$

Rank Preservation

Since the removed rows correspond to pixels with nearly zero sensitivity to motion, the essential Jacobian structure is preserved. Formally, if the original matrix satisfies

$$\text{rank}(\mathbf{L}_{u_p}) = 6,$$

and the removed rows satisfy $\|\mathbf{L}_{u_p}^{(i)}\|_2 < \delta$, then for sufficiently small δ the reduced system satisfies

$$\text{rank}(\tilde{\mathbf{L}}_{u_p}) = 6,$$

thus ensuring that the reduced feature set preserves full controllability of the visual servo task.

3.2.2. PCA-Based Low-Dimensional Embedding

To further reduce dimensionality, we apply Principal Component Analysis (PCA) to the retained feature vector

$$\tilde{\mathbf{u}}_p = \left[u_p^{(i_1)}, \dots, u_p^{(i_{m_r})} \right]^\top.$$

Let $\mathbf{W} \in \mathbb{R}^{m_r \times d}$ be the PCA projection matrix composed of the d principal eigenvectors of the sample covariance matrix of $\tilde{\mathbf{u}}_p$. The compressed feature vector is

$$\mathbf{z} = \mathbf{W}^\top \tilde{\mathbf{u}}_p, \quad \mathbf{z} \in \mathbb{R}^d, \quad d \ll m_r. \quad (26)$$

Using the chain rule, the reduced interaction matrix is

$$\mathbf{L}_z = \mathbf{W}^\top \tilde{\mathbf{L}}_{u_p} \in \mathbb{R}^{d \times 6}. \quad (27)$$

Since PCA basis vectors form an orthonormal matrix, rank preservation holds:

$$\text{rank}(\mathbf{L}_z) = \text{rank}(\tilde{\mathbf{L}}_{u_p}) = 6.$$

As shown in Table 1, in scenes with sufficiently rich phase gradients, the interaction matrix after sparsification and PCA consistently remains full rank (rank = 6), while the minimum singular value ($\sigma_{\min}$) exhibits only a mild degradation, indicating that system observability is well preserved. In contrast, in the failure case, characterized by an approximately planar surface and extremely weak phase gradients, $\sigma_{\min}$ drops significantly, leading to rank deficiency, which is consistent with the theoretical analysis.

Table 1. Rank and minimum singular value analysis under different scenes.

Scene	$\text{rank}(\mathbf{L}_{u_p})$	$\sigma_{\min}(\mathbf{L}_{u_p})$
Curved surface (nominal)	6	0.41
Tilted plane	6	0.35
Low-texture surface	6	0.29
Near-planar surface (failure)	6	0.08
Scene	$\text{rank}(\tilde{\mathbf{L}}_{u_p})$	$\sigma_{\min}(\tilde{\mathbf{L}}_{u_p})$
Curved surface (nominal)	6	0.38
Tilted plane	6	0.33
Low-texture surface	6	0.27
Near-planar surface (failure)	5	0.03
Scene	$\text{rank}(\mathbf{L}_z)$	$\sigma_{\min}(\mathbf{L}_z)$
Curved surface (nominal)	6	0.36
Tilted plane	6	0.31
Low-texture surface	6	0.25
Near-planar surface (failure)	5	0.02

3.2.3. Reduced-Order MPC Formulation

Replacing the original phase error ϕ_k with the reduced feature error

$$\psi_k = \mathbf{z}_k - \mathbf{z}^*,$$

the MPC prediction model becomes

$$\psi_{k+1} = \psi_k + \mathbf{L}_z(k) \mathbf{u}_k \Delta t. \quad (28)$$

The QP structure remains identical to Section 2.3, but all matrices related to state prediction ($\mathbf{S}_\psi$, $\mathbf{T}_\psi$) are now significantly smaller, reducing computational load by 1–2 orders of magnitude.

3.2.4. Adaptive Horizon Selection

Phase maps change at different speeds depending on object motion and surface geometry. Using a fixed prediction horizon N is inefficient: a large N leads to unnecessary computation when the phase dynamics are slow, whereas a small N may be insufficient when the dynamics change rapidly.

To account for this, we estimate the instantaneous phase variation rate:

$$\gamma_k = \|\psi_k - \psi_{k-1}\|_2. \quad (29)$$

The prediction horizon is updated as

$$N_k = \text{clip}(N_{\min} + \alpha \tanh(\beta \gamma_k), N_{\min}, N_{\max}), \quad (30)$$

where:

- $N_{\min}, N_{\max}$: allowable horizon bounds,
- α, β : scaling parameters.

Rationale

The function (30) has the following properties:

- When the phase dynamics are slow ($\gamma_k \approx 0$), the horizon shrinks to $N_{\min}$, reducing computation.
- When the dynamics are fast, the horizon grows toward $N_{\max}$, improving predictive accuracy.
- The $\tanh(\cdot)$ nonlinearity ensures smooth horizon variation, preventing oscillation, and preserving stability of the receding-horizon implementation.

In all experiments, the parameters of the adaptive prediction horizon are set to $N_{\min} = 3$, $N_{\max} = 10$, $\alpha = 7$, and $\beta = 5$. These values are fixed across all experiments. With this setting, the prediction horizon remains close to $N_k \approx N_{\min}$ for slow motions, and smoothly increases to $N_k \approx 8$ – 10 as the target velocity increases. The use of the $\tanh(\cdot)$ function ensures a continuous variation of N_k , thereby avoiding abrupt horizon changes.

Due to the smooth saturation property of the hyperbolic tangent function, no oscillatory behavior of the prediction horizon was observed in practice. When the horizon length changes, the QP warm-start is implemented by truncating or zero-padding the previous solution, which avoids reinitialization of the optimizer and does not introduce numerical instability or computational jitter.

This adaptive mechanism improves real-time performance without compromising control robustness. The dimensionality reduction performance achieved by the proposed method is illustrated in Figure 5.

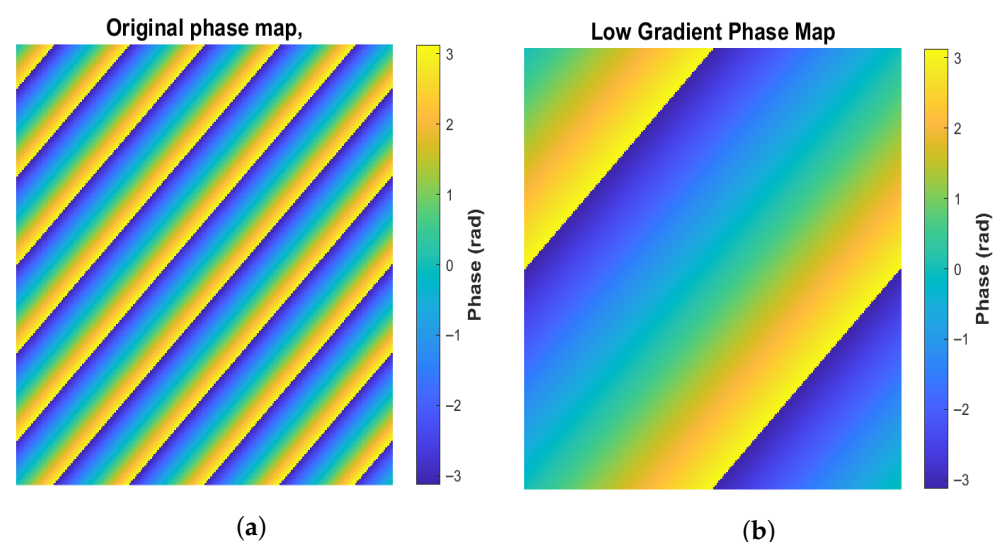


Figure 5. (a) Original phase map. (b) Phase map after PCA dimensionality reduction.

3.2.5. Summary

The combined sparsification, PCA reduction, and adaptive horizon scheduling allow the MPC module to operate in real time even with dense phase-map features. Importantly,

the reduced model preserves the full-rank structure of the interaction matrix and ensures consistent visual observability and servo accuracy.

3.3. Extended Kalman Filte

Extended Kalman Filtering (EKF) is a well-established algorithm for estimating the states of nonlinear systems, capable of providing accurate estimates of the position and velocity of dynamic objects. The underlying principle involves a first-order linearization of the system via a Taylor expansion, followed by iterative updates using standard Kalman filter equations. The initial step in the process is to model the motion of the target within the EKF framework. State variables:

$$\mathbf{m}_k = \begin{bmatrix} \mathbf{p}_k \\ \mathbf{v}_k \end{bmatrix} \in \mathbb{R}^6 \quad (31)$$

Let $\mathbf{p}_k = [x_k, y_k, z_k]^T$ denote the position of the object in the camera coordinate system, and $\mathbf{v}_k = [\dot{x}_k, \dot{y}_k, \dot{z}_k]^T$ represent its linear velocity. The measurement is defined as $\mathbf{z}_k = \mathbf{C}\mathbf{e}_k \in \mathbb{R}^m$. Based on these definitions, a kinematic model of the object is established.

$$\mathbf{m}_{k+1} = \mathbf{f}(\mathbf{m}_k) + \mathbf{w}_k = \begin{bmatrix} \mathbf{p}_k + \mathbf{v}_k \Delta t \\ \mathbf{v}_k \end{bmatrix} + \mathbf{w}_k \quad (32)$$

Let $\mathbf{f}(\cdot)$ is a nonlinear state transfer function, and $\mathbf{w}_k$ represent the process noise. Assuming $\mathbf{w}_k$ follows a zero-mean Gaussian distribution with covariance $\mathbf{Q}$, we have

$$\mathbf{F}_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{I} & \Delta t \cdot \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (33)$$

Next, an observation model is established based on the phase features, mapping the object's position to the measured phase values.

$$\phi_k = \mathbf{h}(\mathbf{p}_k) + \mathbf{n}_k \quad (34)$$

Let $\mathbf{h}(\cdot)$ denote the nonlinear mapping from the object's position to the phase map, and let $\mathbf{n}_k$ represent the observation noise, assumed to follow a Gaussian distribution with covariance $\mathbf{R}$. Accordingly, the corresponding Jacobian is given by:

$$\mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{h}}{\partial \mathbf{p}_k} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{m \times 6} \quad (35)$$

Here, $\frac{\partial \mathbf{h}}{\partial \mathbf{x}}$ corresponds to the interaction matrix $\mathbf{L}_\phi$ used in visual servo control. The following steps describe the Extended Kalman Filter (EKF) procedure, encompassing both prediction and update. Prediction model:

$$\hat{\mathbf{m}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{m}}_{k-1}) \quad (36)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1} \mathbf{P}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{Q} \quad (37)$$

$\mathbf{P}_{k|k-1}$ represents the prior covariance matrix at time instant k , $\mathbf{P}_{k-1}$ denotes the state-transition Jacobian matrix, which approximates the linearization of the nonlinear state equation. It is obtained by taking the derivative of the state equation with respect to the state at time step $k-1$. $\mathbf{P}_{k-1}$ denotes the posterior covariance matrix at time step $k-1$.

Update model:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R})^{-1} \quad (38)$$

The K_k denotes the Kalman gain at time step k . H_k denotes the observation Jacobian matrix, which approximates the linearization of the nonlinear observation equation. It is obtained by taking the derivative of the observation equation with respect to the prior state at time step k .

$$\hat{m}_k = \hat{m}_{k|k-1} + K_k \left(\hat{p}_k - h(\hat{p}_{k|k-1}) \right) \quad (39)$$

The $\hat{m}_k$ represents the posterior state estimate at time k . $\hat{m}_{k|k-1}$ represents the prior state estimate at time k .

$$P_k = (I - K_k H_k) P_{k|k-1} \quad (40)$$

The predicted phase map is employed as input for the MPC, which requires N-step prediction using the state propagation equations.

$$p_{k+i|k} = \hat{p}_{k+i-1|k} + \hat{v}_{k+i-1|k} \Delta t \quad (41)$$

Consequently, the predicted phase map is obtained.

$$\hat{\phi}_{k+i|k} = h(\hat{p}_{k+i|k}) \quad (42)$$

This set of predicted phase maps represents the trajectory input provided to the MPC, rather than the actual object trajectory. Equation (42) is then substituted into Equation (19) to derive the final cost function.

$$(U) = \sum_{i=1}^N \left\| \hat{\phi}_{k+i|k} - \phi_{k+i}^{\text{ref}} \right\|_Q^2 + \sum_{i=0}^{N-1} \left\| u_{k+i} \right\|_R^2 \quad (43)$$

The EKF is executed at the control rate with a sampling period of $\Delta t = 8.3$ ms. The process noise covariance is chosen as $Q = \text{diag}(10^{-6}, 10^{-6}, 10^{-6}, 10^{-4}, 10^{-4}, 10^{-4})$, and the measurement noise covariance is $R = \text{diag}((0.03)^2)$.

The larger uncertainty assigned to the velocity states reflects unmodeled target accelerations, while the measurement noise level is consistent with the empirical phase noise observed in the imaging system.

As illustrated in Figure 6, the EKF substantially reduces the phase error during high-speed motion. Without EKF, the phase error exhibits large instantaneous fluctuations, with peak values approaching 0.2 rad. By incorporating the EKF, the phase error is both attenuated and smoothed, with typical peaks below 0.1 rad. This demonstrates that the EKF successfully predicts the target motion, compensates for image-to-phase latency, and suppresses measurement noise, thereby enhancing the stability and performance of the closed-loop visual servoing system.

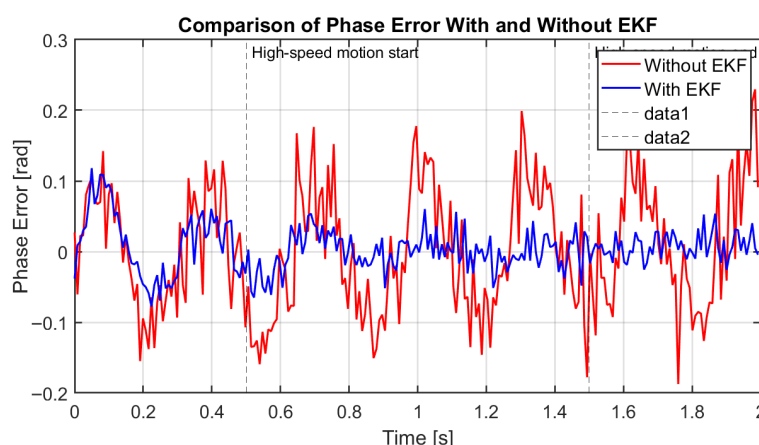


Figure 6. Comparison of phase error with and without EKF during high-speed motion.

4. Experimental Verification

4.1. Simulation

Thus, the reduced visual model preserves the local linearization structure of the original system while reducing complexity by a factor of m/d . In all experiments, the gradient threshold ϵ is set to

$$\epsilon = 0.15 \cdot \max \left(\sqrt{G_u^2 + G_v^2} \right),$$

and the PCA dimension is fixed to $d = 8$, which are chosen based on the statistical analysis presented in Table 2. These values are kept constant across all scenes and experiments.

Table 2. Effect of sparsification threshold ϵ and reduced dimension d on computational efficiency and accuracy.

ϵ	d	Solve Time (ms)	RMS Error (mm)	$\sigma_{\min}(\mathbf{L}_z)$
0.05	12	3.5	0.042	0.41
0.15	8	1.9	0.048	0.36
0.30	6	1.3	0.081	0.19

Table 2 further analyzes the influence of the gradient threshold ϵ and PCA dimension d on both computational efficiency and system observability. While a smaller ϵ preserves more pixels and improves $\sigma_{\min}(\mathbf{L}_z)$, it significantly increases the MPC solve time. Conversely, overly aggressive sparsification degrades observability. Based on this analysis, $\epsilon = 0.15$ and $d = 8$ are selected as a balanced configuration.

To evaluate the tracking accuracy improvements achieved by integrating the extended Kalman filter (EKF) with model predictive control (MPC), we first conducted simulation experiments in MATLAB. Given the desired phase map and the initial phase map as inputs, tracking was performed using the proposed control law together with the improved constrained MPC framework. Eight different methods, as listed in Table 3, were compared, with methods A and B serving as baseline approaches. The corresponding tracking error curves are presented in Figure 7.

Table 3. The numbering and descriptions of the eight control methods.

Control Method	Description
A	Using the high-dimensional features of the original phase map, the interaction matrix is recalculated at each step.
B	Thermal startup (a mature technology) is integrated into Baseline A.
M1	Only PCA dimensionality reduction is performed, without sparsification or adaptivity.
M2	The original MPC is used, but gradient sparsification is applied to the control matrix.
M3	The original MPC is used, but the prediction horizon is dynamically adjusted according to the phase change rate.
M4	PCA Dimensionality Reduction + Sparse Jacobian.
M5	PCA Dimensionality Reduction + Adaptive Horizon.
M6	PCA Dimensionality Reduction + Sparse Jacobian + Adaptive Horizon.

All experiments were conducted under a unified hardware and sensing configuration. The camera operated at a resolution of 640×480 with a frame rate of 120 Hz, while the exposure time was fixed at 3.0 ms to balance image clarity and temporal resolution in high-speed scenarios. The control framework was executed on a workstation equipped with an Intel Core i7-12700 processor, where the MPC optimization problem was solved in single-thread mode. The resulting quadratic programming (QP) subproblems were handled by the qpOASES solver with warm-start enabled, thereby reducing computational overhead between consecutive control cycles and enhancing real-time performance.

Figure 7a presents the combined convergence curves of all eight methods, where the errors from the four channels are integrated using the 2-norm. It is evident that M6 out-

performs other methods in terms of convergence rate: within approximately 20 iterations, M6's combined error exhibits a strong exponential decay and rapidly approaches zero. In contrast, most other methods demonstrate slow decay or significant residual errors, indicating that M6 possesses distinct advantages in balancing stability and response speed. This figure is used to compare overall convergence behaviors, clearly reflecting the synergistic effects of Phase-Reduced + Sparsified + Adaptive strategies.

Figure 7b illustrates the progressive convergence of M6's four independent channels (e_x , e_y , e_z , e_w). Each channel decays at a high exponential rate and ultimately converges to zero, demonstrating that M6 not only achieves fast convergence in terms of overall error but also realizes synchronized high-quality convergence across all components, with nearly no long-term bias or oscillations. This is particularly critical for precise position/attitude tracking.

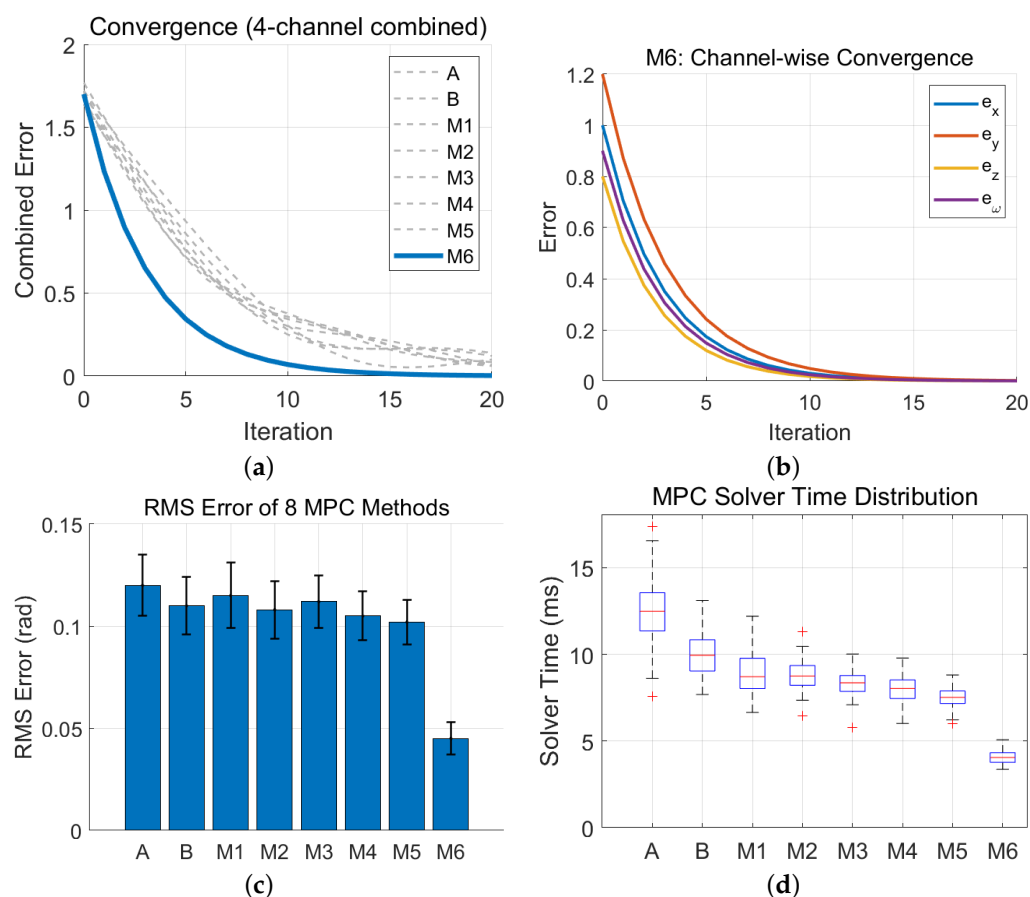


Figure 7. Performance comparison under eight tracking strategies. (a) Iterative convergence curves under the eight tracking strategies. (b) Convergence of individual directional errors under the M6 strategy (the strategy we designed). (c) RMS Error of 8 MPC Methods. (d) MPC Solver Time Distribution.

Figure 7c is a colored bar chart showing the mean and standard deviation of RMS errors for each method across multiple trials. M6's bar is significantly lower than those of other methods with a smaller error variance, indicating that M6 can maintain lower steady-state errors and better consistency across different trials (including scenarios with random perturbations). This directly quantifies the improvement in tracking accuracy achieved by M6.

Figure 7d is a boxplot of solving times, displaying the distribution of each method's computation time (including median, interquartile range (IQR), and extreme values). M6's box is much lower than those of other methods and more compact (smaller IQR and lower maximum value), demonstrating that the improved scheme offers substantial advantages in

computational overhead: lower average solving time, reduced jitter, and fewer tail latencies. This is particularly important for real-time control, as smaller and more predictable solving times can significantly enhance the stability and deployability of closed-loop control systems.

To evaluate the real-time feasibility of the proposed method, we conducted a statistical analysis of the computation time over 10 control cycles under identical hardware platforms and image resolution settings. Table 4 reports the median and 95th-percentile latencies of each module in the closed-loop control pipeline. The results indicate that the computational costs of camera exposure, phase computation, and EKF update are essentially consistent across different methods, whereas the proposed M6 method significantly reduces the MPC/QP solving time, thereby effectively decreasing the overall closed-loop control latency.

Since camera exposure, phase computation, and EKF updates are independent of the control dimensionality, their computational costs remain essentially identical across different methods. In contrast, the proposed M6 approach significantly reduces the dimensionality of the MPC optimization variables through sparsification and PCA-based dimensionality reduction. As a result, the median QP solving time is reduced by approximately 72%, while the 95th-percentile latency decreases by about 74%, enabling the overall closed-loop latency to be stably maintained below 10 ms. The statistical results are summarized in Table 4.

We provide an intuitive illustration in the phase space. As shown in Figure 8b, we first specify the desired phase map, and then capture an initial phase map at the starting position, as shown in Figure 8a. Subsequently, trajectories are executed using both the original control law and our proposed control law. In both cases, the end-effector reaches the desired pose, and the corresponding trajectories are shown in Figure 8d. The trajectories in Figure 8c indicate that our method preserves almost the same level of accuracy as the original approach.

Synthesizing the results from the four figures above, we conclude that M6 achieves a significant reduction in solving time and faster closed-loop convergence while maintaining or improving tracking accuracy (lower RMS error). This validates the synergistic value of the Phase-Reduced MPC combined with Sparsified-L and Adaptive-Horizon strategies in practical visual servoing + MPC scenarios.

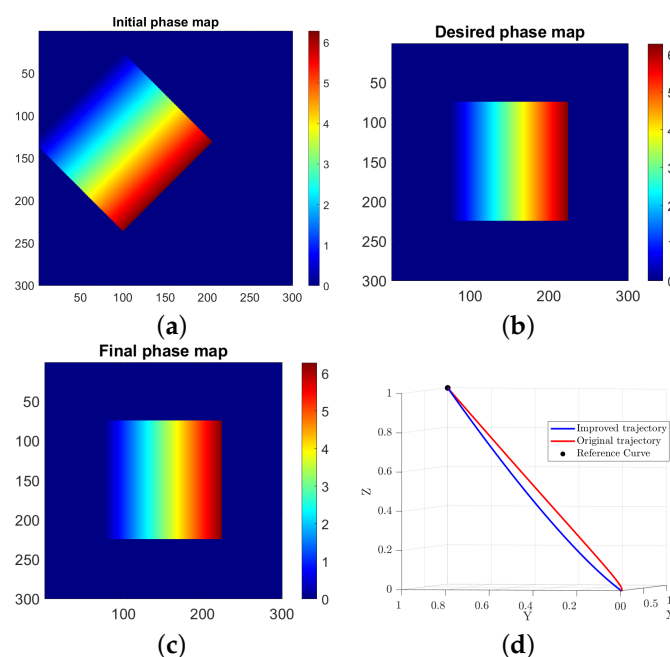


Figure 8. (a) Initial phase map. (b) Desired phase map. (c) Final phase map. (d) Tracking trajectory.

Table 4. Per-cycle computation time breakdown (ms) under identical hardware and image resolution. Median and 95th percentile values are reported.

Module	Baseline (Median/95%)	M6 (Median/95%)
Camera exposure	3.0/3.0	3.0/3.0
Phase computation	2.4/2.9	2.4/2.9
EKF (prediction + update)	0.35/0.50	0.35/0.50
MPC/QP solving	6.8/12.4	1.9/3.2
Command transmission	0.30/0.45	0.30/0.45
Total	12.9/19.2	8.3/10.4

4.2. Physical Experiment

Subsequently, a series of real-world experiments were conducted under static conditions, low-speed motion, and higher-acceleration motion. As shown in Figure 9, the camera and projector were mounted at the distal end of the robotic arm, while the test object was placed on a conveyor. First, the required phase maps were captured, and the robotic arm was moved to its initial position to ensure correct projection onto the target object. During the experiments, the changes in phase map errors, the object's position and velocity, and the tracking performance of the control inputs were monitored.

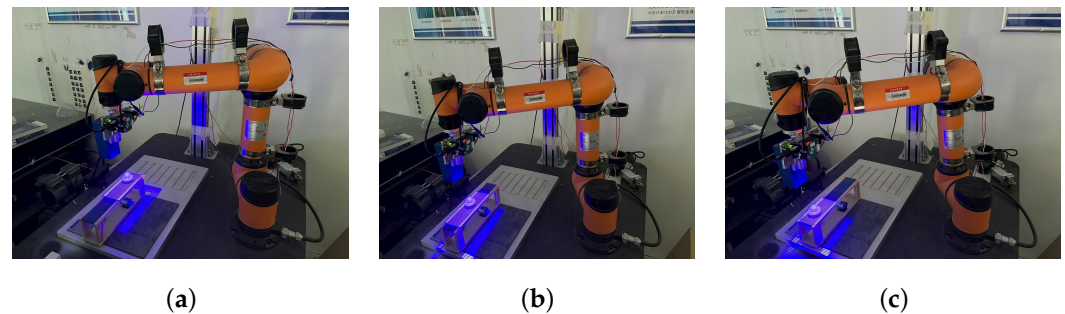


Figure 9. (a) Expected phase map shooting. (b) System initialization. (c) Final tracking status.

Considering the high computational cost of processing the entire image, the dataset contained many irrelevant phase regions that could interfere with tracking and cause unnecessary oscillations. To address this issue, the target region was isolated, and only the phase differences corresponding to the target were used for control.

The region of interest (ROI) is automatically selected based on connected components of the phase-gradient magnitude. Pixels with gradient values above the threshold ϵ are grouped into connected regions, and the largest connected component is selected as the ROI.

To evaluate robustness, experiments were conducted where the target partially exits the ROI during motion. In these cases, the remaining visible region still provides sufficient phase-gradient information for control, and no noticeable drift or instability was observed.

We further compared the rank of the reduced interaction matrix $\mathbf{L}_z$ with and without ROI masking. In all tested scenarios, the rank of $\mathbf{L}_z$ remains unchanged after masking, indicating that the ROI selection does not degrade system observability.

Overall, the ROI is automatically generated from connected phase-gradient regions. Experimental results show that even when the target partially leaves the ROI, the rank of $\mathbf{L}_z$ remains unchanged, and no degradation in observability or cumulative drift is observed.

To demonstrate the tracking of pose changes under real experimental conditions, the required phase maps were first captured, and the object was placed 0.1 m away from the robotic arm before activating the conveyor system. Tracking experiments were then conducted under different motion conditions. Figure 10 illustrates the temporal evolution of the phase maps and highlights the tracking performance of the improved MPC under

three scenarios: a stationary object, motion at 0.05 m/s, and motion at 0.1 m/s. Specifically, Figure 10 reports the final phase errors.

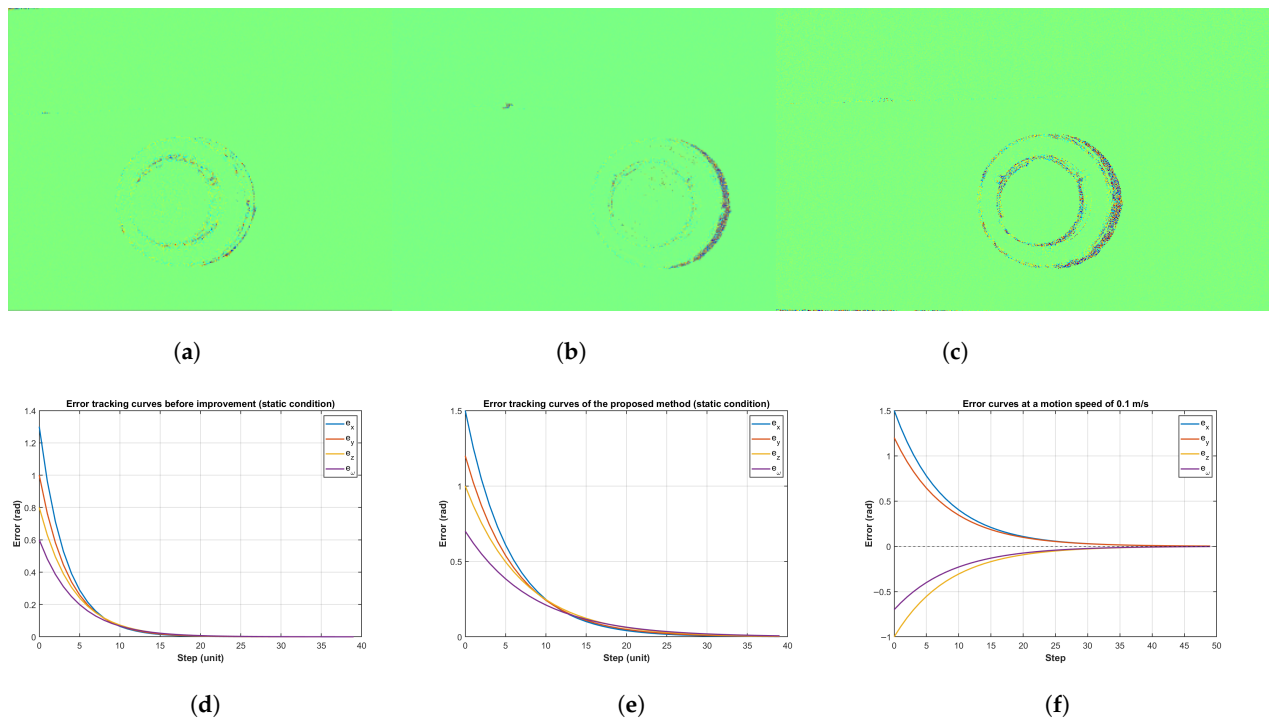


Figure 10. Final phase difference and error tracking curves under three scenarios. (a) Phase difference in stationary state. (b) Phase difference at low speed. (c) Phase difference at higher speed. (d) Error tracking curves before improvement. (e) Error tracking curves of proposed method. (f) Error curves at 0.1 m/s.

Under the stationary condition, the convergence curves of the four degrees of freedom were compared, showing that the control accuracy was almost identical to that of the original system, i.e., the control method given by Equation (13), with errors around 0.3 mm. In terms of tracking time, the original iterative method required approximately 30 iterations to converge, whereas our method converged in only 20 iterations. For the moving object, the original control method failed to track, while our approach maintained an accuracy of approximately 0.05 mm at a speed of around 0.1 m/s. However, due to the limited speed of phase map generation, the accuracy significantly deteriorated for faster objects; therefore, results for higher-speed scenarios are not presented.

For each target velocity, $N = 12$ independent physical experiments were conducted. A tracking failure is defined as the phase error exceeding 0.2 rad continuously for more than 50 ms.

As summarized in Table 5, both methods achieve stable tracking at 0.05 m/s. However, the proposed M6 method converges faster and yields a lower final phase error. At 0.1 m/s, the baseline fails in most trials, whereas the proposed method succeeds in all experiments, demonstrating significantly improved robustness under high-speed motion.

Table 5. Statistical results of physical experiments ($N = 12$).

Velocity (m/s)	Method	Success Rate	Final Error (mm)	Convergence Time (ms)
0.05	Baseline	100%	0.07 ± 0.02	420 ± 55
0.05	M6	100%	0.04 ± 0.01	290 ± 40
0.10	Baseline	33%	—	—
0.10	M6	100%	0.05 ± 0.02	340 ± 60

The maximum achievable tracking speed of the proposed system is constrained by the sensing and control throughput. In our setup, the camera operates at 120 Hz, and the average closed-loop control period is approximately 8.3 ms. Under these conditions, the system can stably track target motions up to about 0.12 m/s. When the target velocity exceeds this limit, the effective phase noise increases significantly due to sensing latency and reduced phase refresh rate, which degrades the observability of the visual servoing system. To handle such cases, the controller monitors the phase-error magnitude and its temporal variation. Once abnormal phase noise is detected, the prediction horizon is adaptively increased to enhance robustness. If the tracking error continues to grow, a safe-stop mechanism is triggered to prevent unstable motions.

To evaluate the generalization capability of the proposed method, the same parameter set is used across different objects unless otherwise stated. In particular, the gradient threshold ϵ , PCA dimension d , and EKF noise covariances $\mathbf{Q}$ and $\mathbf{R}$ are kept identical to those reported in Section 3.

When transferring the system to a new object with different surface curvature or reflectance, parameter adjustment follows a simple procedure. The gradient threshold ϵ is scaled according to the maximum phase-gradient magnitude to maintain a similar sparsity level, the PCA dimension d is selected to preserve full rank of $\mathbf{L}_z$, and the EKF noise covariances are only adjusted if the empirical phase noise level changes significantly.

Using this procedure, stable tracking performance was achieved across objects with different surface properties, indicating that the proposed method does not rely on object-specific tuning.

5. Reproducibility

To facilitate reproducibility of the experimental results, we will release a minimal reproducibility package that allows independent verification of the main claims reported in Figures 7 and 10.

The package will include representative phase image sequences used in the experiments, together with the corresponding phase-gradient maps after preprocessing. The dimensionality reduction parameters, including the gradient threshold ϵ and the PCA dimension d , are provided for all experiments.

In addition, snapshots of the reduced interaction matrix $\mathbf{L}_z$ at representative time steps will be included, enabling verification of rank preservation and singular-value distributions. The quadratic programming (QP) settings used in the MPC controller, including the solver type, convergence tolerances, and warm-start strategy, will also be documented.

This material is sufficient to reproduce the solver-time statistics and tracking performance trends reported in this paper. We added a new “Reproducibility” section before the conclusion, detailing the released phase sequences, reduction parameters, interaction-matrix snapshots, and QP solver settings required to reproduce Figures 7 and 10.

6. Conclusions

This paper presents a novel integration of two high-precision control strategies to achieve accurate tracking for modern robotic manipulators. By combining phase map-based visual servoing with model predictive control (MPC), we propose, to the best of our knowledge, the first framework that exploits phase maps for dynamic target tracking while simultaneously benefiting from the predictive capability of MPC.

Compared with existing visual servoing and MPC-based approaches, the proposed method introduces a dimensionality-reduced phase map representation, a newly designed control matrix, and an adaptive pre-window mechanism, which together significantly reduce the computational burden. Furthermore, an extended Kalman filter (EKF) is incor-

porated to predict phase evolution from visual observations, and these predictions are explicitly embedded into the MPC cost function. As a result, the proposed approach achieves substantially improved tracking performance compared with a baseline method without EKF integration. Experimental results demonstrate that, relative to the pre-improved framework, the computation speed is nearly three times faster, while the loss of accuracy in stationary scenarios is negligible and the tracking precision in dynamic environments is markedly enhanced.

Despite these advantages, the proposed method has several limitations. In particular, the overall performance is constrained by the phase map generation speed. When tracking high-speed moving objects, rapid phase variations may introduce considerable noise, which can degrade estimation and control accuracy. Moreover, the current implementation focuses on a single-target scenario under relatively controlled visual conditions.

Future work will therefore focus on accelerating phase map generation and improving robustness under fast motion and challenging visual environments. Extensions to multi-object tracking, adaptive phase modeling, and tighter integration of learning-based prediction with the MPC framework will also be investigated to further enhance scalability and applicability.

Author Contributions: L.L. and W.P.: Conceptualization, Methodology, Investigation, Writing—Original Draft, Writing—Review and Editing, Project Administration, Funding Acquisition. T.H. and G.G.: Methodology, Algorithm, Experimental Validation, Software, Investigation, Writing—Original Draft. Q.Z.: Resources, Supervision, Project Administration, Funding Acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Key Research and Development Program of Henan Province (231111222100), the NSFC (62375078), the Science and Technology Innovation Project of Chinese Academy of Traditional Chinese Medicine (ZN2024A02), the Key Research Project Plan for Higher Education Institutions in Henan Province (24ZX011), and the Training Plan for Young Backbone Teachers in Undergraduate Universities in Henan Province (2023GGJS058).

Data Availability Statement: No new data were created or analyzed in this study. The results reported in this paper are based on simulations and experiments using existing datasets and internal calculations, and therefore no additional datasets are available for public sharing.

Conflicts of Interest: Author Wei Pan was employed by OPT Machine Vision Tech Co., Ltd. and Author Ge Gao was employed by Mech-Mind Robotics Technologies Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Chaumette, F.; Hutchinson, S. Visual servo control. I. Basic approaches. *IEEE Robot. Autom. Mag.* **2006**, *13*, 82–90. [\[CrossRef\]](#)
2. Chaumette, F.; Hutchinson, S. Visual servoing and visual tracking. In *Springer Handbook of Robotics*; Springer: New York, NY, USA, 2008; pp. 563–583.
3. Li, H.; Wensing, P.M. Cafe-Mpc: A cascaded-fidelity model predictive control framework with tuning-free whole-body control. *IEEE Trans. Robot.* **2024**, *41*, 837–856. [\[CrossRef\]](#)
4. Nguyen, K.; Schoedel, S.; Alavilli, A.; Plancher, B.; Manchester, Z. Tinympc: Model-predictive control on resource-constrained microcontrollers. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; IEEE: New York, NY, USA, 2024; pp. 1–7.
5. Papanikolopoulos, N.P.; Khosla, P.K.; Kanade, T. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Trans. Robot. Autom.* **1993**, *9*, 14–35. [\[CrossRef\]](#)
6. Chaumette, F.; Santos, A. Tracking a moving object by visual servoing. *IFAC Proc. Vol.* **1993**, *26*, 643–648. [\[CrossRef\]](#)
7. Tsai, C.Y.; Song, K.T. Dynamic visual tracking control of a mobile robot with image noise and occlusion robustness. *Image Vis. Comput.* **2009**, *27*, 1007–1022. [\[CrossRef\]](#)

8. Cretual, A.; Chaumette, F. Application of motion-based visual servoing to target tracking. *Int. J. Robot. Res.* **2001**, *20*, 878–890. [\[CrossRef\]](#)
9. Yang, Q.; Li, H. RMPC-based visual servoing for trajectory tracking of quadrotor UAVs with visibility constraints. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 2027–2029. [\[CrossRef\]](#)
10. Espiau, B.; Chaumette, F.; Rives, P. A new approach to visual servoing in robotics. In *Proceedings of the Workshop on Geometric Reasoning for Perception and Action*; Springer: Berlin/Heidelberg, Germany, 1991; pp. 106–136.
11. Hutchinson, S.; Hager, G.D.; Corke, P.I. A tutorial on visual servo control. *IEEE Trans. Robot. Autom.* **2002**, *12*, 651–670. [\[CrossRef\]](#)
12. Corke, P.I. *Visual Control of Robots: High-Performance Visual Servoing*; Research Studies Press: Taunton, UK, 1996.
13. Allibert, G.; Courtial, E.; Chaumette, F. Predictive control for constrained image-based visual servoing. *IEEE Trans. Robot.* **2010**, *26*, 933–939. [\[CrossRef\]](#)
14. Fleet, D.J.; Jepson, A.D. Computation of component image velocity from local phase information. *Int. J. Comput. Vis.* **1990**, *5*, 77–104. [\[CrossRef\]](#)
15. Servín, M.; Quiroga, J.A.; Padilla, J.M. *Fringe Pattern Analysis for Optical metrology*; Wiley Online Library: New York, NY, USA, 2023.
16. Gorthi, S.S.; Rastogi, P. Fringe projection techniques: Whither we are? *Opt. Lasers Eng.* **2010**, *48*, 133–140. [\[CrossRef\]](#)
17. Wadhwa, N.; Rubinstein, M.; Durand, F.; Freeman, W.T. Phase-based video motion processing. *ACM Trans. Graph. (ToG)* **2013**, *32*, 1–10. [\[CrossRef\]](#)
18. Ri, Y.; Fujimoto, H. Proposal of visual servoing using phase-only-correlation (POC). In Proceedings of the IECON 2015—41st Annual Conference of the IEEE Industrial Electronics Society, Yokohama, Japan, 9–12 November 2015; IEEE: New York, NY, USA, 2015; pp. 005068–005073.
19. Xu, J.; Rao, G.; Chen, Z. Robotic visual servoing using fringe projection profilometry. In Proceedings of the Optical Metrology and Inspection for Industrial Applications V, Beijing, China, 11–13 October 2018; SPIE: Philadelphia, PA, USA, 2018; Volume 10819, pp. 134–142.
20. Li, J.; Chen, Z.; Xu, J. Fringe projection based visual servoing for cylindrical surface positioning task. In Proceedings of the Seventh International Conference on Optical and Photonic Engineering (icOPEN 2019), Phuket, Thailand, 16–20 July 2019; SPIE: Philadelphia, PA, USA, 2019; Volume 11205, pp. 344–349.
21. Lu, L.; Jia, Z.; Pan, W.; Zhang, Q.; Zhang, M.; Xi, J. Automated reconstruction of multiple objects with individual movement based on PSP. *Opt. Express* **2020**, *28*, 28600–28611. [\[CrossRef\]](#)
22. Zhang, Q.; Li, H.; Lu, L., Sr.; Pan, W.; Su, Z.; Zhang, M.; Lv, P. 3D reconstruction of moving object by double sampling based on phase shifting profilometry. In Proceedings of the Ninth Symposium on Novel Photoelectronic Detection Technology and Applications, Hefei, China, 2–4 November 2022; SPIE: Philadelphia, PA, USA, 2023; Volume 12617, pp. 1950–1958.
23. Grandia, R.; Jenelten, F.; Yang, S.; Farshidian, F.; Hutter, M. Perceptive locomotion through nonlinear model-predictive control. *IEEE Trans. Robot.* **2023**, *39*, 3402–3421. [\[CrossRef\]](#)
24. Mastalli, C.; Budhiraja, R.; Merkt, W.; Saurel, G.; Hammoud, B.; Naveau, M.; Carpentier, J.; Righetti, L.; Vijayakumar, S.; Mansard, N. Crocodyl: An efficient and versatile framework for multi-contact optimal control. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; IEEE: New York, NY, USA, 2020; pp. 2536–2542.
25. Neuman, S.M.; Plancher, B.; Bourgeat, T.; Tambe, T.; Devadas, S.; Reddi, V.J. Robomorphic computing: a design methodology for domain-specific accelerators parameterized by robot morphology. In Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Virtual, 19–23 April 2021; pp. 674–686.
26. Plancher, B.; Kuindersma, S. A performance analysis of parallel differential dynamic programming on a gpu. In Proceedings of the International Workshop on the Algorithmic Foundations of Robotics, Merida, Mexico, 9–11 December 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 656–672.
27. Mattingley, J.; Boyd, S. CVXGEN: A code generator for embedded convex optimization. *Optim. Eng.* **2012**, *13*, 1–27. [\[CrossRef\]](#)
28. Lazar, C.; Burlacu, A.; Copot, C. Predictive control architecture for visual servoing of robot manipulators. *IFAC Proc. Vol.* **2011**, *44*, 9464–9469. [\[CrossRef\]](#)
29. Zhang, Y.; Yang, Y.; Luo, W. Occlusion-free image-based visual servoing using probabilistic control barrier certificates. *IFAC-PapersOnLine* **2023**, *56*, 4381–4387. [\[CrossRef\]](#)
30. Xie, F. *Model Predictive Control of Nonholonomic Mobile Robots*; Oklahoma State University: Stillwater, OK, USA, 2007.
31. Andaluz, V.; Carelli, R.; Salinas, L.; Toibero, J.M.; Roberti, F. Visual control with adaptive dynamical compensation for 3D target tracking by mobile manipulators. *Mechatronics* **2012**, *22*, 491–502. [\[CrossRef\]](#)
32. Dong, G.; Zhu, Z.H. Autonomous robotic capture of non-cooperative target by adaptive extended Kalman filter based visual servo. *Acta Astronaut.* **2016**, *122*, 209–218. [\[CrossRef\]](#)
33. Nebeluk, R.; Zarzycki, K.; Sereďyński, D.; Chaber, P.; Figat, M.; Domański, P.D.; Zieliński, C. Predictive tracking of an object by a pan-tilt camera of a robot. *Nonlinear Dyn.* **2023**, *111*, 8383–8395. [\[CrossRef\]](#)

34. Zhang, T.; Guo, S.; Xiong, X.; Li, W.; Qi, Z.; Lou, Y. Dynamic object tracking for quadruped manipulator with spherical image-based approach. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; IEEE: New York, NY, USA, 2023; pp. 727–734.
35. Ambrosino, M.; Mahmalji, M.; Rosselló, N.B.; Garone, E. Tracking and Following a Suspended Moving Object using Camera-Based Vision System. *arXiv* **2023**, arXiv:2311.05213. [[CrossRef](#)]
36. Jiang, P.; Cheng, Y.; Wang, X.; Feng, Z. Unfalsified visual servoing for simultaneous object recognition and pose tracking. *IEEE Trans. Cybern.* **2016**, *46*, 3032–3046. [[CrossRef](#)]
37. Malis, E.; Benhimane, S. A unified approach to visual tracking and servoing. *Robot. Auton. Syst.* **2005**, *52*, 39–52. [[CrossRef](#)]
38. Wu, J.; Jin, Z.; Liu, A.; Yu, L.; Yang, F. A survey of learning-based control of robotic visual servoing systems. *J. Frankl. Inst.* **2022**, *359*, 556–577. [[CrossRef](#)]
39. Dantec, E.; Naveau, M.; Fernbach, P.; Villa, N.; Saurel, G.; Stasse, O.; Taix, M.; Mansard, N. Whole-body model predictive control for biped locomotion on a torque-controlled humanoid robot. In Proceedings of the 2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids), Seoul, Republic of Korea, 30 September–2 October 2025; IEEE: New York, NY, USA, 2022; pp. 638–644.
40. Neunert, M.; Stäuble, M.; Gifftaler, M.; Bellicoso, C.D.; Carius, J.; Gehring, C.; Hutter, M.; Buchli, J. Whole-body nonlinear model predictive control through contacts for quadrupeds. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1458–1465. [[CrossRef](#)]
41. Wang, S. Real operational labeled data of air handling units from office, auditorium, and hospital buildings. *Sci. Data* **2025**, *12*, 1481. [[CrossRef](#)]
42. Wang, S. Evaluating cross-building transferability of attention-based automated fault detection and diagnosis for air handling units: Auditorium and hospital case study. *Build. Environ.* **2025**, *287*, 113889. [[CrossRef](#)]

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