IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

SLOT-MPC: A Hierarchical Whole-Body Model Predictive Controller to Enhance Simultaneous Localization and Object Tracking for UAVs

Zhengyu Hua, Li Xing, Yiwei Wu, Can Wang, *Senior Member, IEEE*, Wencan Lu, and Haoyao Chen, *Senior Member, IEEE*

***Abstract*—This paper proposes *SLOT-MPC*, a hierarchical model predictive control framework for a system of multirotor Unmanned Aerial Vehicle (UAV), which aims to minimize uncertainty in estimating both ego-motion and a moving object, thus enhancing the performance of Simultaneous Localization and Object Tracking (SLOT). The framework consists of two layers: OT-MPC (Object Tracking-Model Predictive Controller) for high-level path planning, and a full model whole-body SL-MPC (Self Localization-Model Predictive Controller) for path tracking and view control. The OT-MPC uses a point-mass model with a proposed time-continuous information filter to minimize object estimation uncertainty and computes optimal chasing paths online in a receding horizon manner. Subsequently, to improve visual-based self-localization, the SL-MPC is developed to track the path generated by the OT-MPC, while simultaneously optimizing perception objectives considering observed features to improve visual-based localization. Thus, optimal control sequences for the aerial vehicle are obtained in real time. Experiments are performed to validate the practicability of our approach. We will release our implementation as an open source package for the community.**

***Index Terms*—Aerial systems: Perception and autonomy, whole-body motion planning and control, view planning for SLAM.**

I. Introduction

UNSCREWED Aerial Vehicles (UAVs), equipped with vision sensors, are widely used in industry, due to their agility and compact design. These UAVs serve predominantly in the domains of monitoring [1], surveillance [2], and cinematography [3], wherein autonomous object tracking and chasing [4] assume paramount significance. During the execution of an object tracking and chasing mission, possessing precise localization information of both the object and the UAV itself becomes indispensable.

Accurate self-localization of UAVs using onboard sensors is crucial for ensuring safety and practicality when performing tasks such as aerial object tracking in unknown scenarios. Specifically, the visual Simultaneous Localization and Mapping (vSLAM) technique [5], [6] has enabled UAVs to effectively localize themselves in unknown and GPS-denied environments. To address challenges in dynamic environments that include moving entities, researchers have developed Simultaneous Localization and Object Tracking (SLOT) [7] as an advanced form of SLAM. SLOT integrates moving object detection and tracking capabilities into SLAM operations, providing a viable solution to this problem.

To achieve SLOT, the entire problem can be decoupled into a classical SLAM process and the object tracking process, classified as loosely-coupled solutions. Due to its computational efficiency, loosely-coupled SLOT has been implemented on resource-limited platforms like UAVs [8]. In [7], a decoupled solution is first proposed, utilizing camera pose and object tracking estimators within a filter-based framework. In [9], a VIO approach is used for loosely-coupled SLOT, achieving accurate 3D object tracking without shape priors through monocular image-based object scale recovery. A recent study, Dynamic-VINS [10], introduces an RGB-D inertial odometry system designed for dynamic environments, which enables real-time operation on resource-constrained platforms while incorporating dynamic detection and VIO capabilities. However, in environments such as snowfield or water surface, where the classical SLAM process is prone to failure due to factors such as a lack of features, loosely-coupled SLOT including its object tracking process cannot work properly [11].

In order to enhance the performance of robots in perception-aware tasks such as self-localization or object tracking, researchers have started investigating the generation of robot motion to optimize information acquisition. This includes maximizing area coverage and improving estimation accuracy, known as the Active Information Acquisition (AIA) problem [12], a highly active area of research. The objective of improving SLAM and object tracking performance can be both addressed as AIA problems, which are referred to as **active SLAM** [13], [14] and **active object tracking** [15], respectively. Active object

Received 7 June 2025; accepted 9 July 2025. Date of publication 23 July 2025; date of current version 20 August 2025. This article was recommended for publication by Associate Editor G. Shi and Editor G. Loianno upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant U21A20119 and Grant U1713206 and in part by Shenzhen Science and Innovation Committee Funds under Grant RCJC20231211090050082, Grant SZXJP20230703093206015, and Grant JCYJ20241202123714019. *(Corresponding authors: Can Wang; Haoyao Chen.)*

Zhengyu Hua, Li Xing, Yiwei Wu, and Haoyao Chen are with the School of Intelligence Science and Engineering, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: hychen5@hit.edu.cn).

Can Wang is with the CAS Key Laboratory of Human-Machine Intelligence-Synergic Systems, Shenzhen Institutes of Advanced Technology, Shenzhen 518055, China (e-mail: can.wang@siat.ac.cn).

Wencan Lu is with the Department of Spine Surgery, Shenzhen University General Hospital, Shenzhen 518055, China (e-mail: spine_sugh@szu.edu.cn).

The source code associated with this work can be found on git: https://github.com/HITSZ-NRSL/SLOT-MPC.

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3592059, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3592059

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

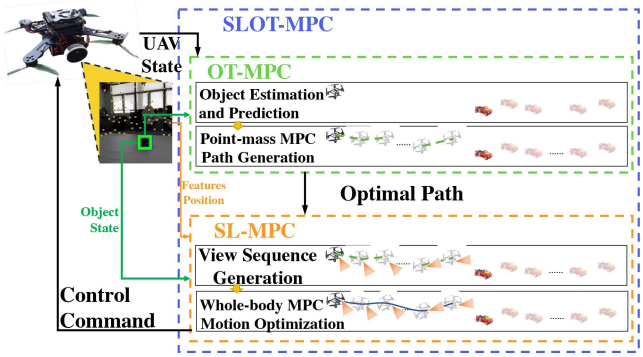

Fig. 1. SLOT-MPC: A hierarchical framework consists of point-mass OT-MPC and whole-body SL-MPC.

tracking can be defined as the task of planning the trajectory of a sensing agent to minimize estimation uncertainty in the state of a specific dynamic object. The problem is generally formulated as an Optimal Control Problem (OCP), while Model Predictive Control (MPC) presents a viable solution for tackling these active object tracking problems in real time [16]. A perception-aware MPC [17] is proposed for quadrotor-UAVs, optimizing robust sensing by maximizing point of interest visibility and minimizing its velocity in the image plane. In recent works such as [15], NMPC approaches with perception constraints have been proposed for generic aerial vehicles, which integrate Kalman Filter for object tracking and aim to minimize uncertainty of object estimation.

Active SLAM can be described as an enhanced version of active object tracking, in which the object is comprised of multiple features within the environment. Therefore, similar to the aforementioned active object tracking solutions, active SLAM can also be tackled using MPC [18], [19]. Since the SLOT task includes both SLAM and object tracking tasks, the design of MPC should further consider object-aware objectives and constraints, such as ensuring the object remains consistently in view.

In summary, there are already existing AIA methods to improve SLAM and object tracking capabilities, respectively. However, currently there are no methods that unify active SLAM and active object tracking, which ultimately optimizing the effectiveness of SLOT. In this paper, our objective is to bridge the gap between active SLAM and active object tracking, thus improving SLOT performance. By formulating the SLOT task as an AIA problem to minimize uncertainty in self-localization and object estimation, we develop an approach that actively plans and controls the UAV along with its gimbal sensor, named as *SLOT-MPC*. To the best of the authors' knowledge, this is the first approach that couples action and perception to enhance the performance of both self-localization and object tracking tasks simultaneously.

The main contribution of this paper is three-fold:

- *OT-MPC:* An innovative approach that integrates the information filter into the MPC framework, leveraging the predictive capabilities of MPC to optimize object estimation accuracy in tracking tasks.
- *SL-MPC:* A sampling-based view planning algorithm aims at ensuring visual-localization accuracy and object visibility, followed by a whole-body NMPC for motion optimization and control. Unlike traditional MPC, this approach introduces a unique sampling-optimization structure that incorporates environmental awareness. This structure avoids local optima and eliminates the need to solve complex OCP. This innovative structure significantly boosts visual localization accuracy by over 50% in featureless environments and effectively mitigates the risk of localization failure.
- *SLOT-MPC:* A systematic, flexible and efficient hierarchical whole-body motion planning and control framework combines the above-mentioned two modules, which simultaneously enhances self-localization and object estimation for UAVs. This two-level design emphasizes the priority of object tracking, and manages to avoid the local optima trap. The architecture of SLOT-MPC is illustrated in Fig. 1.

II. System Overview

A. Modeling

We first analyze the dynamic model of the agent: a multirotor aerial vehicle. The configuration of this vehicle is determined by the 3D localization of the center of mass $\mathbf{p}_b^W$ and the rotation matrix from the body to the world frame $\mathbf{R}_b^W$. For the sake of brevity, we denote $\mathbf{p}_b^W$ by $\mathbf{p}_b$. The following equations describe the system dynamics:

$$\begin{cases} \dot{\mathbf{p}}_b = \mathbf{v}_b \\ \dot{\mathbf{R}}_b^W = \mathbf{R}_b^W \hat{\boldsymbol{\omega}}_b \\ m\dot{\mathbf{v}}_b = m\mathbf{a}_b = -mg\mathbf{e}_3 + f_u \mathbf{R}_b^W \mathbf{e}_3 \\ \mathbf{J}\dot{\boldsymbol{\omega}}_b = -\boldsymbol{\omega}_b \times \mathbf{J}\omega_b + \mathbf{M}_u, \end{cases} \tag{1}$$

where the unit vector $\mathbf{e}_3 = [0,0,1]^T$; $\mathbf{v}_b = \mathbf{v}_b^W$ and $\mathbf{a}_b = \mathbf{a}_b^W$ are the velocity and acceleration of the center of mass expressed in the world frame, respectively; g is the acceleration of gravity; m is the total mass, and $\mathbf{J}$ is the inertia matrix with respect to the body frame; the angular velocity in the body-fixed frame is represented as $\boldsymbol{\omega}_b$ and $\hat{\boldsymbol{\omega}}_b$ is the skew-symmetric matrix associated with $\boldsymbol{\omega}_b$; f_u and $\mathbf{M}_u = [M_{roll}, M_{pitch}, M_{yaw}]^T$ are the control thrust and moment generated by the propellers, respectively.

Due to the inherent underactuation of multirotor aerial vehicle in most cases, their equipped sensors are frequently mounted on gimbals to enhance their flexibility in attitude. Thus, the camera state $\mathbf{x}_C$ in world frame has the following relation with the vehicle body pose:

$$\begin{cases} \mathbf{x}_C = [\mathbf{p}_C^T, \mathbf{q}_C^T]^T \\ \mathbf{p}_C = \mathbf{p}_b \\ \mathbf{q}_C = q(\mathbf{R}_C^W) \\ \mathbf{R}_C^W = \mathbf{R}_b^W \mathbf{R}_C^b(\boldsymbol{\theta_g}) \\ \mathbf{c} = \mathbf{c}_C^W = \mathbf{R}_C^W \mathbf{e}_1 = \mathbf{R}_C^W [1,0,0]^T. \end{cases} \tag{2}$$

Without loss of generality, we make the approximation that the 3D position of the camera in the world frame, denoted as $\mathbf{p}_C$, is equal to that of the UAV body $\mathbf{p}_b$; $q : \mathbf{SO}(3) \to \mathbb{H}$ represents a mapping relation that transform a 3D rotation matrix to quaternion; the camera attitude in body frame $\mathbf{R}_C^b$ is a function of gimbal joint angles $\boldsymbol{\theta_g} \in \mathbb{R}^{n_g}$, where n_g denotes the number of joints; the camera's view direction in world frame is represented by a 3D unit vector $\mathbf{c}_C^W$. In this paper $\mathbf{c}_C^W$ is denoted by $\mathbf{c}$ for brevity. Fig. 2 provides an overview about the reference frames.

In the SLOT task, the primary mission is object tracking, which involves maintaining an appropriate camera pose with

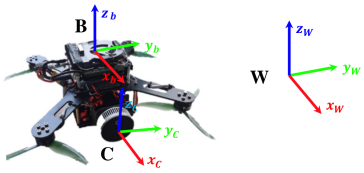

Fig. 2. A schematic diagram showing the world frame $\mathbf{W}$, the body frame $\mathbf{B}$ and the camera frame $\mathbf{C}$.

respect to a specific moving object. Subsequently, based on this object tracking task, efforts should be made to enhance the self-localization accuracy under the object-in-view constraint. The uncertainty associated with the measurement of a feature (either an object or an environmental landmark) located at $\mathbf{p}_T$ can be quantified using the Fisher Information Matrix (FIM) [20], written as a function of camera state and feature position:

$$\mathbf{M}(\mathbf{x}_C, \mathbf{p}_T) = \mathbf{H}^T \mathbf{V}^{-1} \mathbf{H}, \tag{3}$$

where $\mathbf{H} = \partial \mathbf{z} / \partial \mathbf{p}_T$ represents the Jacobian matrix of the measurement $\mathbf{z}$, and $\mathbf{V}$ is the covariance matrix of sensor noise. Both $\mathbf{z}$ and $\mathbf{V}$ will be elaborated in Section III-A.

As demonstrated in [20], the Fisher Information Matrix (FIM) retains rotation invariance when sensor visibility bounds are excluded. This leads to the simplification $\mathbf{M}(\mathbf{x}_C, \mathbf{p}_T) = \mathbf{M}(\mathbf{p}_C, \mathbf{p}_T)$ in (3), implying that the precision of object measurement is purely a function of the relative camera-object geometry.

B. SLOT-MPC Overview

The SLOT-MPC framework is shown in Fig. 1. The problem's high nonlinearity (especially with a multirotor-UAV dynamic model) increases computational complexity, motivating a two-level architecture.

In the first layer (OT-MPC, detailed in Section III), we focus on object estimation, generating a position trajectory to reduce object estimation uncertainty. Here, due to the FIM's rotation-invariant property, camera states are planned with only translation information (ignoring visibility and rotation). Using $\mathbf{p}_C = \mathbf{p}_b$, we generate an initial UAV body-position sequence (a path).

When a high-level path is planned, SL-MPC (detailed in Section IV) enables low-level path tracking control and view planning. It formulates a whole-body OCP considering path tracking, object visibility, and visual localization quality. By solving this OCP, the optimal SLOT-performance control input sequence is obtained.

III. OT-MPC Design

In this section, we first concentrate on the object tracking mission and introduce the proposed OT-MPC. This MPC approach utilizes a point-mass model to efficiently generate $\{\mathbf{p}^*_{b,k}\}_{k=1}^N$, which represents the optimal path of the UAV body for object tracking. The primary goal is to enhance the accuracy of object estimation. This path will be further used as a reference for low-level planning and control.

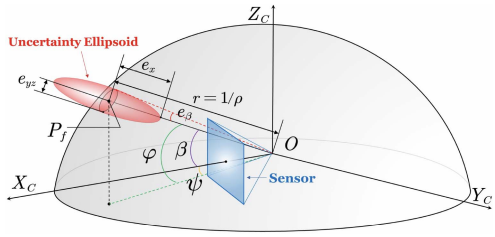

Fig. 3. A range-bearing sensor measures an object at P_f, yielding a measurement of $[\rho, \psi, \varphi]^T$.

A. Object Motion and Observation Model

We denote the state of the observed object T as $\mathbf{x}_T$. Moreover, we model it using a linear Gaussian system:

$$\dot{\mathbf{x}}_T = \mathbf{A}\mathbf{x}_T + \boldsymbol{w}, \quad \boldsymbol{w} \sim \mathcal{N}(0, \mathbf{Q}), \tag{4}$$

where $\mathbf{A}$ represents the state system matrix; $\boldsymbol{w}$ stands for the Gaussian process noise with covariance matrix $\mathbf{Q}$.

When a camera is treated as a range-bearing sensor observing a feature, it measures the distance and direction between the feature and the camera. This measurement, denoted as $[\rho_T, \psi_T, \varphi_T]^T$, represents the relative inverse-depth, azimuth, and elevation. Such a measurement can be converted from spherical coordinates to Cartesian coordinates. The measurement $\mathbf{z}$ and its Jacobian $\mathbf{H}$ are calculated respectively as:

$$\mathbf{z} := \begin{bmatrix} x_T^C \\ y_T^C \\ z_T^C \end{bmatrix} + \boldsymbol{v} = \begin{bmatrix} \rho_T^{-1} \\ \cos\psi_T \cos\varphi_T \\ \rho_T^{-1} \sin\psi_T \cos\varphi_T \\ \rho_T^{-1} \sin\varphi_T \end{bmatrix} + \boldsymbol{v}$$

$$= \mathbf{R}_W^C(\mathbf{p}_T - \mathbf{p}_C) + \boldsymbol{v}, \boldsymbol{v} \sim \mathcal{N}(0, \mathbf{V}) \tag{5a}$$

$$\mathbf{H} = \partial \mathbf{z} / \partial \mathbf{p}_T = \mathbf{R}_W^C, \tag{5b}$$

where $[x_T^C, y_T^C, z_T^C]^T$ represents the 3D position of the object in camera frame; $\mathbf{R}_W^C$ represents the rotation from the world frame to the camera frame, which can be obtained from the UAV and gimbal state according to (2); $\mathbf{p}_T$ and $\mathbf{p}_C$ denote the 3D positions of the object and UAV body in world frame, respectively; $\boldsymbol{v}$ is the Gaussian measurement noise with covariance matrix $\mathbf{V}$.

Fig. 3 shows the measurement covariance matrix $\mathbf{V}$ visualized as a red ellipsoid centered at the measurement mean. The incident angle from the object to the camera is designated as β_T. This angle is also the one between the bearing vector $\overrightarrow{OP_f}$ and the camera's optical axis X_C. Additionally, e_ρ and e_β stand for the variance of the measurement error in inverse-depth and bearing angle, respectively. The eigenvalues of an error covariance matrix are indicative of the variances of errors in different directions. To be specific, regarding to the covariance matrix $\mathbf{V}$, its eigenvalue associated with the range $\overrightarrow{OP_f}$ can be calculated as $e_x = e_r = \rho_T^{-2} e_\rho$, while the eigenvalues associated with the bearing can be computed as $e_{yz} = \rho_T^{-1} e_\beta$. Consequently, the covariance matrix of measurement for the object is expressed as:

$$\mathbf{V} = \mathbf{V}(\mathbf{p}_T, \mathbf{p}_C) = \mathbf{R}_\beta \, diag(\rho_T^{-2} e_\rho, \rho_T^{-1} e_\beta, \rho_T^{-1} e_\beta) \mathbf{R}_\beta^T, \tag{6}$$

where e_ρ and e_β are sensor-specific values; $\mathbf{R}_\beta = \mathbf{R}_z(\psi)\mathbf{R}_y(\varphi)$ represents the rotation between the camera optical axis X_C and

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

the bearing vector $\overrightarrow{OP_f}$. The corresponding measurement information matrix $\mathbf{M}(\mathbf{p}_C, \mathbf{p}_T)$ is now computed based on (3), (5b) and (6). This relationship between measurement information matrix and the UAV-object mutual pose lies in its potential for MPC to exploit and reduce estimation uncertainty.

B. Object State Estimation

We propose a time-continuous version of information filter for object state estimation based on the plant defined by (3), (4) and (5a). This filter is written as:

$$\begin{aligned}\dot{\hat{\xi}} &= -(\mathbf{A}^T + \mathbf{SQ})\xi + \mathbf{H}^T\mathbf{V}^{-1}\mathbf{z} \\ \dot{\mathbf{S}} &= -(\mathbf{SA} + \mathbf{A}^T\mathbf{S} + \mathbf{SQS}) + \mathbf{H}^T\mathbf{V}^{-1}\mathbf{H} \\ &= -(\mathbf{SA} + \mathbf{A}^T\mathbf{S} + \mathbf{SQS}) + \mathbf{M}(\mathbf{p}_C, \mathbf{p}_T),\end{aligned} \tag{7}$$

where $\mathbf{S} = \mathbf{P}^{-1}$ represents the information matrix, the inverse of the estimation covariance matrix $\mathbf{P}$; $\hat{\xi} = \mathbf{S}\mathbf{x}_T$ represents the information vector; $\mathbf{M}(\mathbf{p}_C, \mathbf{p}_T)$ is the FIM mentioned in (3). The derivation of (7) is detailed in the Appendix A.

C. MPC With Information State

We define the object tracking task as an optimal control problem (OCP), with the main objective of reducing estimation uncertainty. To combine the UAV body state and the estimation uncertainty of the object in the MPC framework, we choose to use a static motion model for computational efficiency. This simplifies the object state representation to only include position state, denoted as $\mathbf{x}_T = [\mathbf{p}_T] \in \mathbb{R}^3$. Consequently, the system matrix $\mathbf{A} = \mathbf{0}_3$, aiming to reduce complexity in the system. By modeling the UAV as a point-mass, the MPC state vector is written as:

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_b^\top & vech(\mathbf{S})^\top\end{bmatrix}^\top \in \mathbb{R}^{12}, \tag{8}$$

where the state of UAV body $\mathbf{x}_b^T := [\mathbf{p}_b^T, \mathbf{v}_b^T]^T$ encompasses its 3D position and velocity; $vech(\cdot) : \mathbb{S}_{\succ 0}^{n_T \times n_T} \to \mathbb{R}^{n_T(n_T+1)/2}$ is a half-vectorization operator applied to a symmetric matrix. As a result, the half-vectorization of a position information matrix $\mathbf{S}$ is written as:

$$vech(\mathbf{S}) = [s_{xx}, s_{xy}, s_{xz}, s_{yy}, s_{yz}, s_{zz}] \in \mathbb{R}^6. \tag{9}$$

To mitigate imprecision caused by the static model, an external Kalman predictor with a more comprehensive motion model involving velocity is introduced. The sequence of its predicted object positions $\{\mathbf{p}_{T,k}^*\}_{k=1}^N$ is then utilized in the OT-MPC to calculate the associated $\mathbf{V}_k$ through (6), predicting how the information matrix $\mathbf{S}$ would evolve.

We model the UAV body as a point mass and define its acceleration as the control input, which is described by the dynamic equation f_{dyn}, given by:

$$\mathbf{u} := \mathbf{a}_b, \dot{\mathbf{x}}_b = f_{dyn}(\mathbf{x}_b, \mathbf{u}) := [\mathbf{v}_b, \mathbf{u}]^T. \tag{10}$$

To reduce the estimation uncertainty, we choose to minimize the negative of the natural logarithm for the trace of $\mathbf{S}$, denoted as $-log\,\mathrm{tr}(\mathbf{S})$, as it is computationally efficient and negative correlated with estimation accuracy.

To ensure the system stability, energy and velocity cost are introduced as square Euclidean norms, weighted by diagonal matrices $\mathbf{W}_u$ and $\mathbf{W}_v$, written as $J_{u,k} = \mathbf{u}_k^T\mathbf{W}_u\mathbf{u}_k$ and $J_{v,k} = \mathbf{v}_{b,k}^T\mathbf{W}_v\mathbf{v}_{b,k}$, respectively. We further define the safety constraint $d_{\min} \le d_{b,T,k} \le d_{\max}$. This constraint ensures that the UAV remains at a distance within the interval $[d_{\min}, d_{\max}]$ from the object at all times, where $d_{b,T,k}$ denotes the distance between the object and the UAV at time step k.

The discrete-time OCP in OT-MPC, sampled at N shooting points over the receding horizon $\tau = N\Delta t$ at a given instant t, is expressed as:

$$\min_{\substack{\mathbf{x}_0 \dots \mathbf{x}_N \\ \mathbf{u}_0 \dots \mathbf{u}_{N-1}}} \sum_{k=0}^{N} \left(-w_p \log \mathrm{tr}\,(\mathbf{S}_k) + J_{v,k}\right) + \sum_{k=0}^{N-1} J_{u,k} \tag{11}$$

$$\begin{aligned}\text{s.t.}\quad & \mathbf{x}_0 = \mathbf{x}(t) \\ & \mathbf{x}_{k+1} = \mathbf{f}_{pm}\left(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_{T,k}^*\right) \\ & \mathrm{tr}\,(\mathbf{S}_k) = s_{xx,k} + s_{yy,k} + s_{zz,k} \\ & \mathbf{v}_{\min} \le \mathbf{v}_{b,k} \le \mathbf{v}_{\max} \\ & d_{\min} \le d_{b,T,k} \le d_{\max} \\ & \mathbf{u}_{\min} \le \mathbf{u}_k \le \mathbf{u}_{\max},\end{aligned}$$

where $\mathbf{x}(t)$ denotes the current state defined in (8), whose transition $\mathbf{x}_{k+1} = \mathbf{f}_{pm}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_{T,k}^*)$ is a combination of (7) and (10).

The externally predicted object path $\{\mathbf{p}_{T,k}^*\}_{k=1}^N$ is provided to OT-MPC as an external parameter. Using the control input sequence $\{\mathbf{u}_k^*\}_{k=0}^{N-1}$ obtained from OT-MPC, an optimal path $\{\mathbf{p}_{b,k}^*\}_{k=1}^N$ for the UAV state is then planned. This path is crucial for optimal object tracking and provides a reference for the low-level SL-MPC process, as discussed in the following section.

IV. Self-Localization MPC Design

In section III, we introduce OT-MPC to generate an optimal path $\{\mathbf{p}_{T,k}^*\}_{k=1}^N$ for UAV object tracking and estimation. In this section, we introduce SL-MPC, a model-based approach that empowers the UAV to track the reference path furnished by OT-MPC. Simultaneously, SL-MPC enables the UAV to plan its view in a manner that ensure the quality of visual localization. Compared with [13], [14], which both solely plan view sequences using sampling-based approaches, the proposed SL-MPC adopts a sampling-optimization strategy that combines planning and control with respect to motion and perception objectives. Similar to the objective of OT-MPC in enhancing the estimation quality of a moving object, the view planning problem can also be approached as an AIA problem.

A. View Sequence Generation

Inspired by [13] and [14], we employ a sampling-based method to search for a camera view sequence $\{\mathbf{c}_k^*\}_{k=1}^N$, where the view vector $\mathbf{c}$ is defined in (2). This sequence functions as guidance for the whole-body optimization in Section IV-B and avoids local optima. The generated view sequence yields satisfactory self-localization accuracy based on visual features and object visibility along the optimal path $\{\mathbf{p}_{b,k}^*\}_{k=1}^N$ (already generated by OT-MPC). The COFEE (Covariance-based Feature Exploration-Exploitation) model in [14] is slightly modified to quantify the influence of environmental features on visual localization accuracy, formulated as a weighted sum of information

matrices:

$$\mathbf{I}(\mathbf{c}_k, \mathbf{p}_{b,k}, \mathbf{p}^i_{f,0}, \mathbf{p}_{T,k}) = \sum_i \mathbf{vis}(\beta_{b,k}(\mathbf{p}^i_{f,0}))\mathbf{M}(\mathbf{p}_{b,k}, \mathbf{p}^i_{f,0}) + w_T \mathbf{vis}(\beta_{b,k}(\mathbf{p}_{T,k}))\mathbf{M}(\mathbf{p}_{b,k}, \mathbf{p}_{T,k}), \tag{12}$$

where $\mathbf{p}^i_{f,0}$ denotes the position of i-th feature at current time, which is provided by the baseline visual estimator [21] in real time; $\mathbf{M}(\cdot,\cdot)$ is the FIM mentioned in (3); the mapping $\beta_{b,k}(\cdot) : \mathbb{R}^3 \to \mathbb{R}$ specifies the incident angle between a 3D spatial coordinate $\mathbf{p} \in \mathbb{R}^3$, defined as:

$$\beta_{b,k}(\mathbf{p}) = \arccos \frac{\mathbf{c}_k \cdot (\mathbf{p} - \mathbf{p}_{b,k})}{||\mathbf{p} - \mathbf{p}_{b,k}||}. \tag{13}$$

The weight of the object-related term w_T is set to a rather large value to ensure that the object is within the view. The visibility function $\mathbf{vis}(\cdot)$ is approximately 1 when β is within the field-of-view, and the function's value progressively approaches zero as β falls outside the FOV, given as:

$$\mathbf{vis}\,(\beta) = 1/(1 + e^{-a\cos\left(\frac{\pi\beta}{2f_{ov}}\right)}), \tag{14}$$

where a is a constant. We use the trace of $\mathbf{I}$ defined in (12) as a metric to quantify the information acquisition. The view-sequence generation problem is then defined as searching for a series of camera views $\{\mathbf{c}^*_k\}^N_{k=1}$ that satisfy the following information requirements:

$$\begin{gathered} tr(\mathbf{I}_k) = tr(\mathbf{I}(\mathbf{c}_k, \mathbf{p}_{b,k}, \mathbf{p}^i_{f,0}, \mathbf{p}_{T,k})) > \mathbf{trace}_{\min} \\ \text{for } k = 1, 2 \dots N \\ \text{s.t. } (\mathbf{c}_{k+1} \cdot \mathbf{c}_k) > \cos(\delta_{\max}), \end{gathered} \tag{15}$$

where the latest inequality constraint bounds the angle between two consecutive views within the threshold angle $\delta_{\max}$. Similar to [14], this feasible path problem can be efficiently solved by the Time-based RRT algorithm [22].

B. Receding Horizon View Optimization

In the optimization framework of SL-MPC, the robot state $\mathbf{x}_r$ is expressed as the concatenation of the UAV body state $\mathbf{x}_b$ and n_g-DOF gimbal state $\mathbf{x}_g \in \mathbb{R}^{n_g}$, which is defined as:

$$\mathbf{x}_r = [\mathbf{x}^T_b, \mathbf{x}^T_g]^T = [\mathbf{p}^T_b, \mathbf{q}^T_b, \mathbf{v}^T_b, \boldsymbol{\omega}^T_b, \boldsymbol{\theta}^T_g, \boldsymbol{\omega}^T_g]^T \in \mathbb{R}^{13+2n_g}. \tag{16}$$

The velocity of the gimbal joints is denoted as $\boldsymbol{\omega}_g = \dot{\boldsymbol{\theta}}_g$. The acceleration of the gimbal joints, denoted as $\boldsymbol{\alpha}_g = \dot{\boldsymbol{\omega}}_g$, is combined with the control thrust and moments of the UAV body to form the control input for the optimization problem:

$$\mathbf{u}_r = [f_u, \mathbf{M}^T_u, \boldsymbol{\alpha}^T_g]^T \in \mathbb{R}^{4+n_g}. \tag{17}$$

In section IV-A, a reference camera view sequence $\{\mathbf{c}^*_k\}^N_{k=1}$ is obtained, ensuring satisfactory self-localization accuracy and object visibility along the optimal path $\{\mathbf{p}^*_{b,k}\}^N_{k=1}$. The first goal of SL-MPC is to drive $\mathbf{c}_k$, the camera view at time step k, to approach $\mathbf{c}^*_k$. As a result, the corresponding cost function is designed as follows, weighted by a scalar w_{sl}:

$$J_{sl,k} = \mathbf{w}_{sl}(1 - \mathbf{c}_k \cdot \mathbf{c}^*_k). \tag{18}$$

To enable the UAV to smoothly track the path $\{\mathbf{p}^*_{b,k}\}^N_{k=1}$ generated by OT-MPC, we define the reference tracking and energy objectives as quadratic costs weighted by diagonal matrices $\mathbf{W}_{pt}$ and $\mathbf{W}_{ur}$, respectively, written as:

$$\begin{aligned} J_{pt,k} &= (\mathbf{p}_{b,k} - \mathbf{p}^*_{b,k})^T \mathbf{W}_{pt} (\mathbf{p}_{b,k} - \mathbf{p}^*_{b,k}) \\ J_{ur,k} &= \mathbf{u}^T_{r,k} \mathbf{W}_{ur} \mathbf{u}_{r,k}. \end{aligned} \tag{19}$$

Finally, the discrete-time OCP in SL-MPC is written as:

$$\min_{\substack{\mathbf{x}_{r,0} \dots \mathbf{x}_{r,N} \\ \mathbf{u}_{r,0} \dots \mathbf{u}_{r,N-1}}} \sum_{k=0}^{N} (J_{sl,k} + J_{pt,k}) + \sum_{k=0}^{N-1} J_{ur,k} \tag{20}$$

$$\begin{aligned} \text{s.t. } & \mathbf{x}_{r,0} = \mathbf{x}_r(t) \\ & \mathbf{x}_{r,k+1} = \mathbf{f}_r(\mathbf{x}_{r,k}, \mathbf{u}_k) \\ & \boldsymbol{\Omega}_{b,\min} \le \Omega(\mathbf{q}_{b,k}) \le \boldsymbol{\Omega}_{b,\max} \\ & \boldsymbol{\omega}_{b,\min} \le \boldsymbol{\omega}_{b,k} \le \boldsymbol{\omega}_{b,\max} \\ & \boldsymbol{\theta}_{g,\min} \le \boldsymbol{\theta}_{g,k} \le \boldsymbol{\theta}_{g,\max} \\ & \boldsymbol{\omega}_{g,\min} \le \boldsymbol{\omega}_{g,k} \le \boldsymbol{\omega}_{g,\max} \\ & \mathbf{u}_{r,\min} \le \mathbf{u}_{r,k} \le \mathbf{u}_{r,\max}, \end{aligned}$$

where the state transition function $\mathbf{x}_{r,k+1} = \mathbf{f}_r(\mathbf{x}_{r,k}, \mathbf{u}_k)$ is obtained from (1), (16), and (17); $\boldsymbol{\Omega}_b = [\phi_b, \theta_b, \psi_b]^T$ denotes the UAV body's Euler angles; $\Omega : \mathbb{H} \to \mathbb{R}^3$ is the mapping from quaternions to Euler angles.

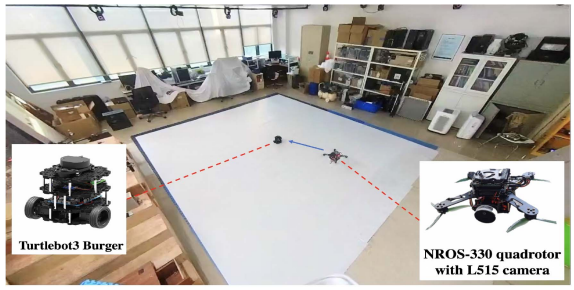

Fig. 4. Experimental setup: an NROS-330 quadrotor UAV with a tiltable camera chases a Turtlebot3 Burger ground mobile robot on a white ground.

V. Experiments

A. Experimental Setup

In both OT-MPC and SL-MPC, the MPC optimization steps are implemented using ACADO, with qpOASES serving as the solver. These implementations can be compiled for a specified middleware, such as ROS. Moreover, a 4^{th} order explicit Runge-Kutta integrator is employed. We selected a discretization step of $dt = 0.1$ s and a time horizon of $\tau = 1.0$ s, executing one iteration per control loop at a frequency of 100 Hz.

The experiment setup is shown in Fig. 4. We use the NROS-330, a compact quadrotor, to conduct the object tracking tasks. The NROS-330 is equipped with a tiltable RGB-D camera (Intel Realsense L515) and an onboard computer of Intel NUC i7-1260P. Dynamic-VINS [21] serves as the baseline SLOT module, detecting moving object in view and enclosing them with bounding box. Features outside the bounding box are used for estimating the camera state $\mathbf{x}_C$, while features inside are utilized to track the pose of the object and derive its measured position $\mathbf{p}^{meas}_T$. The associated information matrix $\mathbf{M}(\mathbf{x}_C, \mathbf{p}^{meas}_T)$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

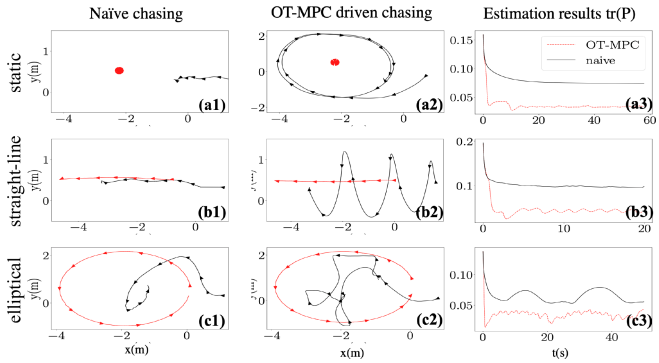

Fig. 5. Proposed OT-MPC vs. naive chasing: Each row illustrates the chasing performance across three motion types. **Columns 1–2**: The (x, y) trajectories of the target (red) and UAV (black) are shown for *Naive Chasing* (a1–c1) and *OT-MPC* (a2–c2). **Column 3 (a3–c3)**: Temporal evolution of the trace of matrix **P** is compared between the two strategies.

TABLE I
OBJECT POSITION ESTIMATION: OT-MPC VS. NAIVE CHASING

	OT-MPC		Naive Chasing	
object motion	mean(tr(**P**))	mean(d)	mean(tr(**P**))	mean(d)
static	3.7e-2	1.54	7.9e-2	1.59
straight-line	4.6e-2	1.57	1.1e-1	1.56
ellipse	3.4e-2	1.74	6.5e-2	1.70

The mean of position estimation covariance **P** (mean(tr(**P**))) and UAV-object distance $d = d_{b,T}$ in 3 cases with different object motion types. mean(tr(**P**)) has units of m^2, while the average distance mean(d) is measured in meters.

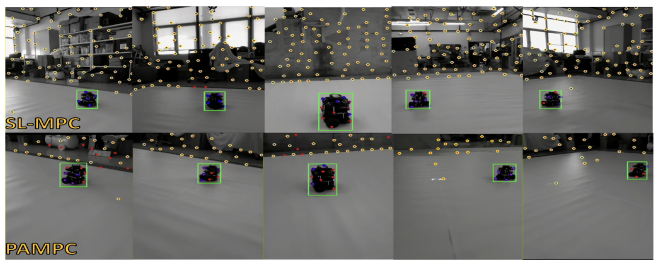

Fig. 6. Snapshots of the camera view using SL-MPC (top row) and PAMPC (bottom row) during SLOT tasks. Any two snapshots in the same column are taken when the UAV and the object are both in similar positions.

is derived through (3), (5a), and (6). And a Turtlebot3 Burger ground mobile robot is employed as a moving object to be chased by the UAV in the experiment. An OptiTrack motion capture system is utilized to obtain the ground-truth pose of both the UAV and the ground mobile object. The ground of the experimental environment is white and featureless, which challenges the UAV in visual-localization, as if it were chasing mobile object on the snowfield.

B. Optimal Object Chasing and Estimation Control

This section presents real-world experimental validation of OT-MPC's uncertainty minimization in its role as the high-level optimal object-tracking controller. Additionally, a naive chasing method, which solely considers the relative pose between the UAV and the object, is implemented by setting $w_p = 0$ in (11). This chasing strategy without uncertainty-minimization task is the same as the point-mass MPC with only the UAV-object distance constraint as proposed in [23], which is used as control group. These two approaches are compared across three scenarios with varying types of object motion: static, straight-line, and elliptical.

For both of the group of OT-MPC and naive chasing, we employ the same low-level controller, PAMPC [17], a perception-aware controller considering both action and perception objectives. In this experiment, PAMPC in each group both tracks the reference path (from OT-MPC or naïve chasing) and maximizes the object's visibility. The evaluation is based on the trace of the object estimation covariance matrix $\mathbf{P}$, which quantifies the total spatial uncertainty of the estimated object position. The evaluation results showcase how the proposed OT-MPC enhances the quality of object estimation, as depicted in Fig. 5 and Table I.

The first and second column in Fig. 5 show the trajectories of both the object and the UAVs. The first column displays the chasing trajectory under the naive strategy [23]. It is evident that in all three cases, the naive chasing strategy causes the UAV to maintain a fixed distance from the object while losing precision in object position estimation, as it consistently observes the object from similar angles.

Unlike naive chasing, OT-MPC dynamically adjusts the observation angle according to the shape of the covariance error ellipsoids (as depicted in Fig. 3). By leveraging more reliable bearing measurement instead, it becomes possible to mitigate the inaccuracies in range measurement that are common in standard range-bearing sensors. The second column of Fig. 5 shows the chasing trajectory based on OT-MPC. All the chasing results show that the UAV successfully minimizes estimation uncertainties by periodically altering its viewing angle on the object, which continuously utilizes more reliable sensor direction to compensate the higher-uncertainty direction in object estimation. The uncertainty minimization processes are visualized and comprehensively illustrated in the attached video. Furthermore, the average UAV-object distances during chasing, as shown in Table I, illustrate that the UAV, when employing the OT-MPC strategy, does not need to significantly approach the object for enhanced estimation accuracy while compromising safety.

C. Self-Localization During Object Chasing

The localization-enhancement capability of the proposed framework, facilitated by SL-MPC, is validated in comparison with PAMPC [17]. Similar to our proposed SL-MPC, PAMPC considers objectives that are related to both perception and reference tracking.

The comparison experiments between the proposed SL-MPC and PAMPC are performed in three different scenarios: 1) a static object is tracked by OT-MPC, 2) an object moves along an elliptical path and is tracked by naive chasing MPC [23], and 3) an object moves along an elliptical path and is tracked by OT-MPC. In each scenario, the high-level path is generated by high-level controller (OT-MPC or naive chasing MPC) in real-time, while low-level controller (SL-MPC or PAMPC) is implemented subsequently to track the path and meanwhile control its view for satisfactory observation.

A couple of snapshots of the camera view in the experiment are presented in Fig. 6. Obviously, when the UAV uses PAMPC (bottom row) as the low-level controller for the SLOT task, because of the featureless ground and the unawareness of visual

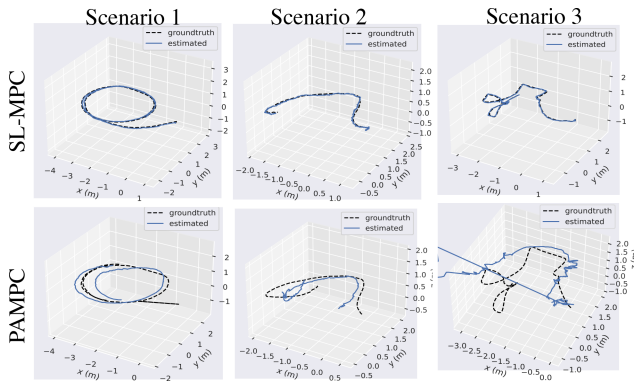

Fig. 7. Visual-based self-localization results: proposed SL-MPC (top) vs. PAMPC (bottom). The blue solid lines and black dashed lines represent the estimated and ground-truth trajectories of UAV, respectively.

TABLE II
SELF LOCALIZATION: SL-MPC VS. PAMPC

	SL-MPC		PAMPC	
Scenario	ATE(m)	RPE(m)	ATE(m)	RPE(m)
1	3.3e-2	5.6e-2	1.9e-1	3.9e-1
2	3.4e-2	6.7e-2	3.3e-1	7.6e-1
3	2.7e-2	5.8e-2	Failed	Failed

features, only a few features are in the view. This situation leads to inaccuracies in self-localization and may even cause it to fail. On the other hand, our proposed SL-MPC benefits from its awareness of both the object and the features in the view. It drives the view towards the feature-rich scene, thus ensuring sufficient localization accuracy and robustness. The self-localization estimation results are illustrated in Fig. 7. Quantitative evaluation results, including Absolute Trajectory Error (ATE) and Relative Pose Error (RPE), are shown in Table II. For additional information, please refer to the attached video.

VI. DISCUSSION

In this section, a qualitative analysis of our approach is presented, along with some comparisons with similar methods in the literature.

A. Design of Hierarchical Framework

To improve the performance of SLOT, we proposed a two-stage framework. In the first stage we solve the OT (Object Tracking) problem, which has the highest priority, and then subsequently solve the SL (Self-Localization) problem in the second stage. Rather than formulating the entire SLOT problem in a nonlinear OCP, which would result in a highly complicated and non-convex objective function, our hierarchical approach emphasizes the priority of object tracking, and manages to avoid the local optima trap.

B. Time-Continuous Information Filter in OT-MPC

We proposed a time-continuous information filter and incorporated it into OT-MPC for estimation uncertainty prediction and minimization. In comparison with a time-discrete information filter, which conducts multiple approximations in information propagation and correction, a time-continuous version is more accurate. Furthermore, the time-continuous information state can be incorporated with the point-mass dynamic model of UAV, which is also time-continuous.

The primary goal of optimal target estimation control is to identify observation configurations that minimize estimation uncertainty. For efficiency, the measurement term $\mathbf{M}(\mathbf{p}_C, \mathbf{p}_T)$ is expected to be incorporated linearly. Unlike the approach of embedding a Kalman filter in MPC [15] which utilizes a nonlinear coupling of the measurement term, the information filter simplifies the measurement update step and adopts a linear incorporation, as shown in (7). This enhances computational efficiency of optimization in MPC.

C. Sampling-Optimization Approach in SL-MPC

In the SL-MPC detailed in Section IV, we first sampled the view sequence that satisfied feature information requirement, and then applied whole-body MPC to drive the view of camera to approach the sampled informative view. Rather than designing an objective function related to feature information in the nonlinear whole-body MPC and then solving it, our method circumvents the local optima trap and does not need to solve a complex OCP with multiple parameters.

VII. CONCLUSION

We propose SLOT-MPC, a hierarchical model predictive framework designed specifically for the two essential tasks of aerial robotics: visual self-localization and object tracking. Through a series of experiments, the efficacy of our approach in producing significant improvements across both visual localization and object tracking tasks is validated.

While SLOT-MPC demonstrates strong performance in typical scenarios, its effectiveness diminishes in extreme cases such as highly dynamic environments with unpredictable moving objects or prolonged occlusion scenarios. Future work will address these limitations through improved prediction models and occlusion-resilient feature tracking. Additionally, we also aim to explore the ways in which this framework can be adapted to deal with situations where a team of agents is involved in a multi-object tracking task.

APPENDIX A
TIME-CONTINUOUS INFORMATION FILTER

Suppose a continuous time-invariant plant is prescribed:

$$\dot{x}(t) = Ax(t) + w(t),\ z(t) = Hx(t) + v(t), \tag{21}$$

with $w(t) \sim \mathcal{N}(0, Q)$, $v(t) \sim \mathcal{N}(0, V)$ and $x(t) \sim \mathcal{N}(\bar{x}(t), P(t))$. If the sampling period Δt is small, the discrete version of (21) is derived using Euler's approximation:

$$x_{k+1} = (I + A_k)x_k + w_k,\ z_k = Hx_k + v_k, \tag{22}$$

with $A_k = A\Delta t$, $w_k \sim \mathcal{N}(0, Q_k) = \mathcal{N}(0, Q\Delta t)$, $v_k \sim \mathcal{N}(0, V_k) = \mathcal{N}(0, V/\Delta t)$ and $x(t) \sim \mathcal{N}(\bar{x}(t), P(t))$. The state and uncertainty can be estimated by information filter:

$$Predict: \quad S_{k+1/k} = \left((I+A_k)S_k^{-1}(I+A_k)^T + Q_k\right)^{-1} \tag{23a}$$

$$\hat{\xi}_{k+1/k} = S_{k+1/k}\,(I + A_k)\,S_k^{-1}\hat{\xi}_k \tag{23b}$$

$$Update: \quad S_{k+1} = S_{k+1/k} + H^T V_k^{-1} H \tag{23c}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

$$\hat{\xi}_{k+1} = H^T V_k^{-1} z_k + \hat{\xi}_{k+1/k}, \tag{23d}$$

where $S = P^{-1}$ represents the information matrix and $\hat{\xi} = P^{-1}x$ represents the information vector. By substituting $A_k = A\Delta t$, $Q_k = Q\Delta t$ and $V_k = V/\Delta t$ into (23a) and (23c), the predict and update steps of the information matrix are written as follows:

$$S_{k+1/k} = ((I + A\Delta t)S_k^{-1}(I + A\Delta t)^T + Q\Delta t)^{-1} \tag{24a}$$

$$S_{k+1} = S_{k+1/k} + H^T V^{-1} H \Delta t. \tag{24b}$$

In the limit as $\Delta t \to 0$, the terms of order Δt^2 are neglected. Based on (24a), the following derivation is obtained:

$$\begin{aligned}\lim_{\Delta t \to 0} S_{k+1/k} &= (S_k^{-1} + (AS_k^{-1} + S_k^{-1}A^T + Q)\Delta t)^{-1} \\ &= (I - S_k(AS_k^{-1} + S_k^{-1}A^T + Q)\Delta t)S_k,\end{aligned} \tag{25}$$

where I stands for an identity matrix. By substituting (25) into (24b) and neglecting terms of order Δt^2, we obtain the rate of change of the information matrix:

$$\begin{aligned}\dot{S} &= \lim_{\Delta t \to 0} (S_{k+1} - S_k)/\Delta t \\ &= -(SA + A^T S + SQS) + H^T V^{-1} H.\end{aligned} \tag{26}$$

By substituting (25) and (26) into (23b) and (23d), the information vector's rate of change over time is derived:

$$\dot{\hat{\xi}} = -(A^T + SQ)\xi + H^T V^{-1} z. \tag{27}$$

REFERENCES

[1] J. A. J. Berni, P. J. Zarco-Tejada, L. Suarez, and E. Fereres, "Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 3, pp. 722–738, Mar. 2009.

[2] H. Huang, A. V. Savkin, and W. Ni, "Online UAV trajectory planning for covert video surveillance of mobile targets," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 2, pp. 735–746, Apr. 2022.

[3] R. Bonatti et al., "Autonomous aerial cinematography in unstructured environments with learned artistic decision-making," *J. Field Robot.*, vol. 37, no. 4, pp. 606–641, 2020.

[4] B. Penin, P. R. Giordano, and F. Chaumette, "Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3725–3732, Oct. 2018.

[5] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8421746/

[6] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proc. 2020 IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4666–4672.

[7] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *Int. J. Robot. Res.*, vol. 26, pp. 889–916, Sep. 2007, doi 10.1177/0278364907081229.

[8] J. Chen, T. Liu, and S. Shen, "Tracking a moving target in cluttered environments using a quadrotor," in *Proc. 2016 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2016, pp. 446–453, doi: 10.1109/iros.2016.7759092.

[9] K. Qiu, T. Qin, W. Gao, and S. Shen, "Tracking 3-D motion of dynamic objects using monocular visual-inertial sensing," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 799–816, Aug. 2019, doi: 10.1109/tro.2019.2909085.

[10] J. Liu, X. Li, Y. Liu, and H. Chen, "RGB-D inertial odometry for a resource-restricted robot in dynamic environments," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9573–9580, Oct. 2022, doi: 10.1109/LRA.2022.3191193.

[11] P. Zhou, Y. Liu, and Z. Meng, "PointSLOT: Real-time simultaneous localization and object tracking for dynamic environment," *IEEE Robot. Automat. Lett.*, vol. 8, no. 5, pp. 2645–2652, May 2023.

[12] H. Jeong, B. Schlotfeldt, H. Hassani, M. Morari, D. D. Lee, and G. J. Pappas, "Learning Q-network for active information acquisition," in *Proc. 2019 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 6822–6827.

[13] Z. Hua, F. Quan, H. Chen, J. Sun, J. Liu, and Y. Liu, "Sampling-based view planning for MAVs in active visual-inertial state estimation," in *Proc. 2022 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 11893–11899.

[14] Z. Hua, B. Xu, F. Quan, J. Sun, and H. Chen, "A covariance-based view planning algorithm for aerial robots with decoupled vision system," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 6, pp. 8419–8430, Dec. 2024.

[15] M. Jacquet and A. Franchi, "Motor and perception constrained NMPC for torque-controlled generic aerial vehicles," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 518–525, Apr. 2021, doi: 10.1109/lra.2020.3045654.

[16] R. Tallamraju et al., "Active perception based formation control for multiple aerial vehicles," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 4491–4498, Oct. 2019, doi: 10.1109/lra.2019.2932570.

[17] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "PAMPC: Perception-aware model predictive control for quadrotors," in *Proc. 2018 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–8.

[18] M. Jacquet and A. Franchi, "Enforcing vision-based localization using perception constrained N-MPC for multi-rotor aerial vehicles," in *Proc. 2022 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 1818–1824.

[19] Z. Wang, H. Chen, S. Zhang, and Y. Lou, "Active view planning for visual SLAM in outdoor environments based on continuous information modeling," *IEEE/ASME Trans. Mechatron.*, vol. 29, no. 1, pp. 237–248, Feb. 2024.

[20] Z. Zhang and D. Scaramuzza, "Beyond point clouds: Fisher information field for active visual localization," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 5986–5992. [Online]. Available: https://ieeexplore.ieee.org/document/8793680/

[21] J. Liu, X. Li, Y. Liu, and H. Chen, "RGB-D inertial odometry for a resource-restricted robot in dynamic environments," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9573–9580, Oct. 2022.

[22] A. Sintov and A. Shapiro, "Time-based RRT algorithm for rendezvous planning of two dynamic systems," in *Proc. 2014 IEEE Int. Conf. Robot. Automat.*, 2014, pp. 6745–6750.

[23] H. Masnavi, V. K. Adajania, K. Kruusamäe, and A. K. Singh, "Real-time multi-convex model predictive control for occlusion-free target tracking with quadrotors," *IEEE Access*, vol. 10, pp. 29009–29031, 2022.