Simultaneous Estimation of Euclidean Distances to a Stationary Object's Features and the Euclidean Trajectory of a Monocular Camera

Zachary I. Bell, *Member, IEEE*, Patryk Deptula, *Member, IEEE*, Emily A. Doucette, J. Willard Curtis, *Member, IEEE*, and Warren E. Dixon, *Fellow, IEEE*

***Abstract*—Data-based, exponentially converging observers are developed for a monocular camera to estimate the Euclidean distance (and hence accurately scaled coordinates) to features on a stationary object and to estimate the Euclidean trajectory taken by the camera while tracking the object, without requiring the typical positive depth constraint. A Lyapunov-based stability analysis shows that the developed observers are exponentially converging without requiring persistence of excitation through the use of a data-based learning method. An experimental study is presented, which compares the developed Euclidean distance observer to previous observers demonstrating the effectiveness of this result.**

***Index Terms*—Computer vision, nonlinear observers, simultaneous localization and mapping (SLAM), structure from motion (SfM), vision-based localization.**

I. Introduction

In many applications, the state (e.g., position and orientation) of an autonomous agent and its local environment (e.g., relative positions of objects in the surrounding environment) must be determined from sensor data. This problem is well known as simultaneous localization and mapping (SLAM) (cf., [1]–[5]). Often, a global positioning system (GPS) is used to estimate the position; however, in many environments, GPS is unavailable (e.g., when agents operate in GPS denied or contested environments) motivating the use of only local sensing data (e.g., camera images, inertial measurement units, and wheel encoders) to estimate the position and model the surrounding environment.

Using cameras to reconstruct the surrounding environment (i.e., determine the Euclidean scale of objects in the environment) requires the assumption that object features are in the camera field of view (FOV) and may be extracted and tracked through a sequence of images. However, a significant challenge arises in determining the scale of objects in an image using a camera given the loss of depth information. Specifically, images of objects are 2-D projections of the 3-D environment. Approaches to reconstruct (i.e., estimate the structure) objects use multiple images of an object along with scale information (cf., [6], [7]) or motion (cf., [8]–[23]), such as linear and angular velocities of the camera. The latter of these methods is referred to as structure from motion (SfM). Generally, the Euclidean scales of objects are not known; however, multiple calibrated cameras may be used to recover the scale (cf., [6], [7]). However, this approach does not work in all scenarios because some objects may have limited or no parallax between the camera images. In SfM approaches, the potential for limited parallax still exists; however, a camera may travel to generate enough parallax, which is generally not possible in stereo vision.

The SfM problem may be approached using online iterative methods (cf., [8]–[24]) and offline batch methods (cf., [6], [7], [25], and the references contained within). These offline approaches perform an optimization over an image sequence, but only show convergence for limited cases (cf., [26], [27]). Most online SfM approaches assume continuous measurements of objects by the camera or only update when a new image is received (cf., [8]–[23]); however, recent results have discussed approaches to handle objects temporarily leaving the camera FOV (cf., [28]–[30]). Many results apply the extended Kalman filter (EKF) to estimate depth (cf., [8], [10]–[12]); however, the EKF generally does not guarantee convergence and may fail in some applications [31], [32]. Compared to the EKF approach, techniques, such as [13], [15], [16], [18], [20], and [22], show asymptotic convergence of the structure estimation errors. Furthermore, results, such as [9], [14], [17], [19], and [21], show exponential convergence of the scale estimate assuming some form of a persistence of excitation (PE) condition or the more strict extended output Jacobian (EOJ) is satisfied. Specifically, the authors in [17] show exponential convergence assuming the PE condition is met and either the initial estimation error is small or the velocities are limited. Furthermore, the development in [19] yields exponential convergence assuming the observer satisfies the EOJ condition. In [21], an exponentially stable observer is developed that requires the motion along at least one axis to be nonzero, and the observer remains ultimately bounded if the PE assumption does not hold, whereas in [19], the observer becomes singular. Typically, SfM approaches require the motion (e.g., linear and angular velocities) to be known; however, the design in [22], extending an approach similar to Dani *et al.* [21], demonstrates a partial solution to the more challenging problem (i.e., compared to SfM) of structure and motion where not only are the feature Euclidean coordinates estimated, but also two of the linear velocities and the three angular velocities of the camera are estimated assuming PE and the linear velocity, and acceleration is measurable along one axis.

In this article, and our preliminary work in [23], exponentially converging observers are developed that use a camera to estimate the Euclidean distance to features on a stationary object in the camera FOV while also estimating the Euclidean trajectory of the camera tracking the object. Unlike previous methods, such as [9], [14], [17], [19], and [21], that assume a PE condition, the developed estimator only requires finite excitation. The finite excitation condition results from the use of concurrent learning (CL) (cf., [33]–[36]). The concept of

Manuscript received July 27, 2020; accepted September 30, 2020. Date of publication November 3, 2020; date of current version August 30, 2021. This work was supported in part by the National Science Foundation under Award 1509516, in part by a Task Order Contract With the Air Force Research Laboratory, Munitions Directorate at Eglin AFB, and in part by AFOSR under Award FA9550-18-1-0109. Recommended by Associate Editor S. K. Spurgeon. *(Corresponding author: Zachary I. Bell.)*

Zachary I. Bell, Patryk Deptula, and Warren E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611-6250 USA (e-mail: bellz121@ufl.edu; pdeptula@ufl.edu; wdixon@ufl.edu).

Emily A. Doucette and J. Willard Curtis are with the Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL 32542 USA (e-mail: emily.doucette@eglin.af.mil; jwillardcurtis@gmail.com).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2020.3035597

CL is to use recorded input and output data from system trajectories to identify uncertain constant parameters of the system in real time under the assumption that the system is sufficiently excited for a finite amount of time. This approach relaxes the PE assumption and can be monitored and verified online. The results in [24] demonstrate exponential convergence of depth estimates using CL; however, these results rely on the assumption that the features are on a plane. CL could be applied to achieve the result in this article; however, motivated by the performance improvement as discussed in [37] (especially for noise-prone image feedback), this article employs integral CL (ICL) (cf., [23], [37]–[39]). ICL removes the necessity to estimate the highest order derivative of the system required in traditional CL (cf., [23], [37]–[39]).

Although ICL removes the need for measuring the state derivative, it still requires the state to be measurable; yet, a unique challenge in this article is that the state depends on the unmeasurable distance to the target. Moreover, the traditional state used in results, such as [8]–[22], [24], includes an inherent singularity when one of the coordinates becomes zero (i.e., the so-called depth to the target). Specifically, previous results assume a positive depth constraint where the distance from the focal point of the camera to the target along the axis perpendicular to the image plane remains positive. The positive depth constraint is satisfied if the features remain in the camera FOV; however, the constraint can be violated for some camera rotations that cause the feature to leave the FOV. Since new results are being developed that allow features to intermittently leave the FOV (cf., [29], [30], [40], [41]), a new formulation of the error system is motivated.

In this article, we exploit alternative image geometry insights to express the error system with a more general distance measure that only becomes zero when the target and camera are coincident; thereby, avoiding the positive depth constraint. While this result also requires the features to remain in the FOV (which ensures that the positive depth constraint is satisfied), eliminating the positive depth constraint eliminates a barrier for future development that would allow intermittent viewing of the features. Although, the new image geometry based error system avoids the potential depth singularity, the resulting error system still contains the unmeasurable distance to the target. However, the development in Section III illustrates how the unmeasurable state can be related to an unknown constant to enable the use of ICL. Regardless of the system identification method used, there is a delay before sufficient excitation occurs to identify the parameters. Therefore, the preliminary result in [23] and the development in Section III exhibit an arbitrarily long delay before determining the feature Euclidean coordinates. In Section IV, we modify the developed learning strategy to include gradient terms that enable transient learning until sufficient data have been collected for the ICL terms.

To illustrate the performance of the developed observers, multiple experiments are presented, including a comparison of the observers in Sections III and IV with the results in [21] and an EKF. These results indicate that the EKF and result in [21] have improved transient performance over the result in Section III, before the ICL-based estimates converge. The EKF and result in [21] have similar transient response as the observer in Section IV before the ICL-based estimates converge. After the ICL-based estimates converge, the observers in Sections III and IV converge to steady state with improved performance over the EKF and observer in [21].

II. Motion Model for Stationary Features

The following definitions and assumptions are presented to aid in the development of the subsequent observers.

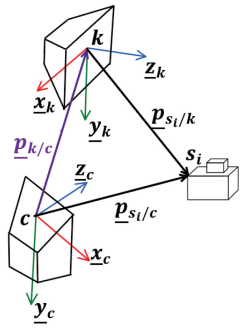

Fig. 1. Example geometry for tracking the position of the ith feature of s. Example shows the camera starts at the top left where the key image is taken and is traveling downward from the upper left to the lower left while tracking a stationary object on the right.

Definition 1: A key frame is defined as the camera frame at which features are first extracted from an image of an object.

The key frame, denoted by $\mathcal{F}_k$, has its origin at the principal point of that image, denoted by k, and basis $\{\underline{x}_k, \underline{y}_k, \underline{z}_k\}$. The frame at which the current image is taken, denoted by $\mathcal{F}_c$, has its origin at the principal point of the current image, denoted by c, and basis $\{\underline{x}_c, \underline{y}_c, \underline{z}_c\}$. This implies that $\mathcal{F}_k$ is established to coincide with $\mathcal{F}_c$ at time $t = 0$.

Assumption 1: There exists a stationary object s with features that can be detected and tracked provided they are within the FOV of the camera. Specifically, for all time $t \in \mathbb{R}_{\geq 0}$, a set of at least $p \in \mathbb{Z}_{\geq 4}$ trackable planar features or $p \in \mathbb{Z}_{\geq 5}$ nonplanar features are on s are in the camera's FOV.

Remark 1: It is assumed that the object remains in the FOV in this result; however, the subsequent development may be extended to not require this constraint. Additionally, using feature extraction techniques, such as [42], a set of features can be extracted from an image of a stationary object. These features can then be tracked while they are in the FOV of the camera using techniques, such as [43] and [44].

Assumption 2: The camera intrinsic matrix $A \in \mathbb{R}^{3\times 3}$ is known and invertable [7].

Assumption 3: The camera linear and angular velocities, $\underline{v}_c(t), \underline{\omega}_c(t) \in \mathbb{R}^3$, are measurable and expressed in $\mathcal{F}_c$.

As shown in Fig. 1, the position of the ith feature on s, $s_i \in \mathbb{Z}_{>0} \quad \forall i = \{1, \ldots, p\}$, can be described as

$$\underline{p}_{s_i/c}(t) = \underline{p}_{k/c}(t) + R_{k/c}(t)\underline{p}_{s_i/k} \tag{1}$$

where $\underline{p}_{k/c}(t) \in \mathbb{R}^3$ is the position of k with respect to c expressed in $\mathcal{F}_c$, $\underline{p}_{s_i/k}(t) \in \mathbb{R}^3$ is the position of feature s_i with respect to k expressed in $\mathcal{F}_k$, $R_{k/c}(t) \in \mathbb{R}^{3\times 3}$ is the rotation matrix describing the orientation of $\mathcal{F}_k$ with respect to $\mathcal{F}_c$, and $\underline{p}_{s_i/c}(t) \in \mathbb{R}^3$ is the position of feature s_i with respect to c expressed in $\mathcal{F}_c$. Rearranging (1) gives

$$\left[\underline{u}_{s_i/c}(t) - \underline{u}_{k/c}(t)\right] \begin{bmatrix} d_{s_i/c}(t) \\ d_{k/c}(t) \end{bmatrix} = R_{k/c}(t)\underline{u}_{s_i/k} d_{s_i/k} \tag{2}$$

where $d_{s_i/c}(t) \in \mathbb{R}_{>0}$ and $\underline{u}_{s_i/c}(t) \in \mathbb{R}^3$ are the distance and unit vector of feature s_i with respect to c expressed in $\mathcal{F}_c$, respectively; $d_{k/c}(t) \in \mathbb{R}_{>0}$ and $\underline{u}_{k/c}(t) \in \mathbb{R}^3$ are the distance and unit vector of k with respect to c expressed in $\mathcal{F}_c$, respectively; and $d_{s_i/k} \in \mathbb{R}_{>0}$ and $\underline{u}_{s_i/k} \in \mathbb{R}^3$ are the distance and unit vector of feature s_i with respect to k expressed in $\mathcal{F}_k$, respectively.

Assumption 4: The origins k and c are not coincident for all time $t > 0$, implying $d_{k/c}(t) > 0$. Additionally, the motion of the camera is

not parallel to the position of a feature, $\|\underline{u}_{k/c}(t) - \underline{u}_{s_i/c}(t)\| > 0$, for all time $t \geq 0$.

Under Assumptions 1–4, the rotation $R_{k/c}(t)$ and unit vector $\underline{u}_{k/c}(t)$ can be determined from a general set of stationary features, using existing techniques, such as planar homography decomposition or essential decomposition.[1] Additionally, $\underline{u}_{s_i/k}$ and $\underline{u}_{s_i/c}(t)$ can always be determined from $\underline{u}_{s_i/k} = \frac{A^{-1}p_{s_i/k}}{\|A^{-1}p_{s_i/k}\|}$ and $\underline{u}_{s_i/c}(t) = \frac{A^{-1}p_{s_i/c}(t)}{\|A^{-1}p_{s_i/c}(t)\|}$, where $p_{s_i/k}$, $p_{s_i/c}(t) \in \mathbb{R}^3$ are the homogeneous pixel coordinates of feature s_i in $\mathcal{F}_k$ and $\mathcal{F}_c$, respectively. Let $H_{s_i}(t) \triangleq [\underline{u}_{s_i/c}(t) \quad -\underline{u}_{k/c}(t)]$. While $d_{k/c}(t) > 0$, (2) is invertable such that

$$\begin{bmatrix} d_{s_i/c}(t) \\ d_{k/c}(t) \end{bmatrix} = \psi_{s_i}(t) d_{s_i/k} \tag{3}$$

where $\psi_{s_i}(t) \triangleq (H_{s_i}^T(t)H_{s_i}(t))^{-1}H_{s_i}^T(t)R_{k/c}(t)\underline{u}_{s_i/k}$ is invertable and measurable under Assumptions 1–4. Furthermore, given $\mathcal{F}_k$ and s are stationary, the time derivatives of the unknown distances are measurable as

$$\frac{d}{dt}\left(d_{s_i/c}(t)\right) = \eta_{s_i,1}(t) \tag{4}$$

$$\frac{d}{dt}\left(d_{k/c}(t)\right) = \eta_{s_i,2}(t) \tag{5}$$

and

$$\frac{d}{dt}(d_{s_i/k}) = 0 \tag{6}$$

where $\eta_{s_i,1}(t)$ and $\eta_{s_i,2}(t)$ are the first and second elements of $\eta_{s_i}(t) \triangleq -\begin{bmatrix} \underline{u}_{s_i/c}^T(t) \\ \underline{u}_{k/c}^T(t) \end{bmatrix}\underline{v}_c(t)$, respectively.

III. Integral CL Observer Update Laws for Euclidean Distances

Motivated by the developments in [23] and [38], an ICL update law is implemented to estimate the constant unknown distances $d_{s_i/k}$ by integrating (4) and (5) over a time window $\varsigma \in \mathbb{R}_{>0}$ yielding

$$\begin{bmatrix} d_{s_i/c}(t) \\ d_{k/c}(t) \end{bmatrix} - \begin{bmatrix} d_{s_i/c}(t-\varsigma) \\ d_{k/c}(t-\varsigma) \end{bmatrix} = \int_{t-\varsigma}^{t} \eta_{s_i}(\iota)\, d\iota, \; t > \varsigma$$

where ς may be constant in size or change over time. While $\int_{t-\varsigma}^{t} \eta_{s_i}(\iota)d\iota$ is a known quantity, $\begin{bmatrix} d_{s_i/c}(t) \\ d_{k/c}(t) \end{bmatrix}$ and $\begin{bmatrix} d_{s_i/c}(t-\varsigma) \\ d_{k/c}(t-\varsigma) \end{bmatrix}$ are unknown; however, the relationship in (3) may be utilized at the current time t and the previous time $t-\varsigma$ yielding

$$\mathcal{Y}_{s_i}(t) d_{s_i/k} = \mathcal{U}_{s_i}(t) \tag{7}$$

where $\mathcal{Y}_{s_i}(t) \triangleq \begin{cases} 0_{2\times 1}, & t \leq \varsigma, \\ (\psi_{s_i}(t) - \psi_{s_i}(t-\varsigma)), & t > \varsigma, \end{cases}$ and $\mathcal{U}_{s_i}(t) \triangleq \begin{cases} 0_{2\times 1}, & t \leq \varsigma \\ \int_{t-\varsigma}^{t} \eta_{s_i}(\iota)d\iota, & t > \varsigma \end{cases}$. The dynamics in (7) demonstrate that CL may be used to estimate the constant distances $d_{s_i/k}$ to the features on s. Specifically, multiplying both sides of (7) by $\mathcal{Y}_{s_i}^T(t)$ yields

$$\mathcal{Y}_{s_i}^T(t)\mathcal{Y}_{s_i}(t) d_{s_i/k} = \mathcal{Y}_{s_i}^T(t)\mathcal{U}_{s_i}(t). \tag{8}$$

In general, $\mathcal{Y}_{s_i}(t)$ will not have full column rank (e.g., when the camera is stationary) implying $\mathcal{Y}_{s_i}^T(t)\mathcal{Y}_{s_i}(t) \geq 0$. However, the equality in (8) may be evaluated at instances in time and summed together (i.e., history stacks) as

$$\Sigma_{\mathcal{Y}_{s_i}} d_{s_i/k} = \Sigma_{\mathcal{U}_{s_i}} \tag{9}$$

where $\Sigma_{\mathcal{Y}_{s_i}} \triangleq \sum_{h_i=1}^{N} \mathcal{Y}_{s_i}^T(t_{h_i})\mathcal{Y}_{s_i}(t_{h_i})$, $\Sigma_{\mathcal{U}_{s_i}} \triangleq \sum_{h_i=1}^{N} \mathcal{Y}_{s_i}^T(t_{h_i})\mathcal{U}_{s_i}(t_{h_i})$, $t_{h_i} \in (\varsigma, t)$, and $N \in \mathbb{Z}_{>1}$. A method for selecting data is subsequently described in Remark 4.

Assumption 5: The camera has sufficiently rich motion so that there exists finite constants $\tau_{s_i} \in \mathbb{R}_{>\varsigma}$, $\lambda_\tau \in \mathbb{R}_{>0}$ such that for all time $t \geq \tau_{s_i}$, $\lambda_{\min}\{\Sigma_{\mathcal{Y}_{s_i}}\} > \lambda_\tau$, where $\lambda_{\min}\{\cdot\}$ and $\lambda_{\max}\{\cdot\}$ are the minimum and maximum eigenvalues of $\{\cdot\}$, respectively.[2] This assumption is an observability condition for the subsequent development that is similar to other image-based observers (cf., [9], [14], [17], [19], [21]); however, if this condition is not satisfied, the observer remains bounded, as demonstrated in the subsequent analysis.

Assumption 5 can be verified online and is heuristically easy to satisfy because it only requires a finite collection of sufficiently exciting $\mathcal{Y}_{s_i}(t)$ and $\mathcal{U}_{s_i}(t)$ to yield $\lambda_{\min}\{\Sigma_{\mathcal{Y}_{s_i}}\} > \lambda_\tau$. The time τ_{s_i} is unknown; however, it can be determined online by checking the minimum eigenvalue of $\Sigma_{\mathcal{Y}_{s_i}}$. After τ_{s_i}, $\lambda_{\min}\{\Sigma_{\mathcal{Y}_{s_i}}\} > \lambda_\tau$ implies that a constant unknown distance $d_{s_i/k}$ can be determined from (9) as

$$d_{s_i/k} = \mathcal{X}_{s_i}, \; t \geq \tau_{s_i} \tag{10}$$

where $\mathcal{X}_{s_i} \triangleq \begin{cases} 0, & t < \tau_{s_i} \\ \Sigma_{\mathcal{Y}_{s_i}}^{-1}\Sigma_{\mathcal{U}_{s_i}}, & t \geq \tau_{s_i} \end{cases}$. When $t \geq \tau_{s_i}$, (10) can be substituted into (3) to yield

$$d_{s_i/c}(t) = \nu_{s_i,1}(t), \; t \geq \tau_{s_i} \tag{11}$$

and

$$d_{k/c}(t) = \nu_{s_i,2}(t), \; t \geq \tau_{s_i} \tag{12}$$

where $\nu_{s_i,1}(t)$ and $\nu_{s_i,2}(t)$ are the first and second elements of $\nu_{s_i}(t) \triangleq \psi_{s_i}(t)\mathcal{X}_{s_i}$, respectively.

The estimation errors, $\tilde{d}_{s_i/c}(t), \tilde{d}_{k/c}(t), \tilde{d}_{s_i/k}(t) \in \mathbb{R}$, are quantified as

$$\tilde{d}_{s_i/c}(t) \triangleq d_{s_i/c}(t) - \hat{d}_{s_i/c}(t) \tag{13}$$

$$\tilde{d}_{k/c}(t) \triangleq d_{k/c}(t) - \hat{d}_{k/c}(t) \tag{14}$$

and

$$\tilde{d}_{s_i/k}(t) \triangleq d_{s_i/k} - \hat{d}_{s_i/k}(t) \tag{15}$$

where $\hat{d}_{s_i/c}(t), \hat{d}_{k/c}(t), \hat{d}_{s_i/k}(t) \in \mathbb{R}$ are the estimates with initial conditions $\hat{d}_{s_i/c}^0, \hat{d}_{k/c}^0, \hat{d}_{s_i/k}^0 \in \mathbb{R}$ selected based on the tracking objective (e.g., user knowledge of the application or related sensors, such as an altimeter, when viewing ground targets from an airborne camera). Motivated by the subsequent stability analysis, the implementable observer update laws for the estimates are designed using (10)–(12) as

$$\frac{d}{dt}\left(\hat{d}_{s_i/c}(t)\right) \triangleq \begin{cases} \eta_{s_i,1}(t), & t < \tau_{s_i} \\ \eta_{s_i,1}(t) + k_1\left(\nu_{s_i,1}(t) - \hat{d}_{s_i/c}(t)\right), & t \geq \tau_{s_i} \end{cases} \tag{16}$$

$$\frac{d}{dt}\left(\hat{d}_{k/c}(t)\right) \triangleq \begin{cases} \eta_{s_i,2}(t), & t < \tau_{s_i} \\ \eta_{s_i,2}(t) + k_2\left(\nu_{s_i,2}(t) - \hat{d}_{k/c}(t)\right), & t \geq \tau_{s_i} \end{cases} \tag{17}$$

[1] See [6], [7], and [45] for examples on calculating the rotation and normalized translation from planar and nonplanar features; however, nonplanar methods will require more than four features to be tracked.

[2] See [46] and [47] for some examples of methods for selecting data to satisfy the assumption.

and

$$\frac{d}{dt}\left(\hat{d}_{s_i/k}(t)\right) \triangleq \begin{cases} 0, & t < \tau_{s_i} \\ k_3\left(\mathcal{X}_{s_i} - \hat{d}_{s_i/k}(t)\right), & t \geq \tau_{s_i} \end{cases} \tag{18}$$

where $k_1, k_2, k_3 \in \mathbb{R}_{>0}$ are constants. Taking the time derivative of (13)–(15), and substituting (10)–(15), (4)–(6), and (16)–(18) yields

$$\frac{d}{dt}\left(\tilde{d}_{s_i/c}(t)\right) = \begin{cases} 0, & t < \tau_{s_i} \\ -k_1\tilde{d}_{s_i/c}(t), & t \geq \tau_{s_i} \end{cases} \tag{19}$$

$$\frac{d}{dt}\left(\tilde{d}_{k/c}(t)\right) = \begin{cases} 0, & t < \tau_{s_i} \\ -k_2\tilde{d}_{k/c}(t), & t \geq \tau_{s_i} \end{cases} \tag{20}$$

and

$$\frac{d}{dt}\left(\tilde{d}_{s_i/k}(t)\right) = \begin{cases} 0, & t < \tau_{s_i} \\ -k_3\tilde{d}_{s_i/k}(t), & t \geq \tau_{s_i} \end{cases} \tag{21}$$

implying for all time $t \geq \tau_{s_i}$, the estimation error derivatives are negative definite functions of the estimation errors. The forms of the update laws in (16)–(18) are implementable and used in practice, whereas the forms of the time derivative of the estimation errors in (19)–(21) are analytical and provided to facilitate the subsequent analysis.

IV. Extended Observer Update Law for Euclidean Distance to Features From Camera

The subsequent analysis demonstrates that (13) and (19) will remain bounded while $t < \tau_{s_i}$. However, after sufficient data are gathered, for all $t \geq \tau_{s_i}$, (13) will be shown to decay exponentially. The delay required to get sufficient excitation may reduce transient performance (i.e., the error is not guaranteed to reduce until after time $t \geq \tau_{s_i}$), which is a disadvantage compared to previous approaches, such as [21], which improve estimation errors by estimating optical flow. Motivated by the optical flow estimator form of the inverse depth estimator in [21], the time rate of change of $\underline{u}_{s_i/c}(t)$ is approximated and used to provide additional information to the estimator in (16), which will improve transient performance while sufficient excitation has not occurred.

Taking the time derivative of $\underline{u}_{s_i/c}(t)$ yields

$$\frac{d}{dt}\left(\underline{u}_{s_i/c}(t)\right) = -\underline{\omega}_c^{\times}(t)\underline{u}_{s_i/c}(t) + \frac{1}{d_{s_i/c}(t)}\left(\underline{u}_{s_i/c}(t)\underline{u}_{s_i/c}^T(t) - I_{3\times 3}\right)\underline{v}_c(t)$$

and

$$\xi_{s_i}^T(t)\xi_{s_i}(t)d_{s_i/c}(t) = \xi_{s_i}^T(t)\rho_{s_i}(t) \tag{22}$$

where $\xi_{s_i}(t) \triangleq (\frac{d}{dt}(\underline{u}_{s_i/c}(t)) + \underline{\omega}_c^{\times}(t)\underline{u}_{s_i/c}(t))$, $\rho_{s_i}(t) \triangleq (\underline{u}_{s_i/c}(t)\underline{u}_{s_i/c}^T(t) - I_{3\times 3})\underline{v}_c(t)$, $\underline{\omega}_c^{\times}(t) \triangleq \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$, and $I_{3\times 3} \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. To aid in the subsequent analysis, let $\mu_{s_i}(t) \triangleq \eta_{s_i,1}(t) + k_\xi(\xi_{s_i}^T(t)\rho_{s_i}(t) - \xi_{s_i}^T(t)\xi_{s_i}(t)\hat{d}_{s_i/c}(t))$, then an extended version of the estimator in (16) is designed as

$$\frac{d}{dt}\left(\hat{d}_{s_i/c}(t)\right) \triangleq \begin{cases} \mu_{s_i}(t), & t < \tau_{s_i} \\ \mu_{s_i}(t) + k_1\left(\nu_{s_i,1}(t) - \hat{d}_{s_i/c}(t)\right), & t \geq \tau_{s_i} \end{cases} \tag{23}$$

where $k_\xi \in \mathbb{R}_{>0}$. Using (13) and (22) in (23), then simplifying yields

$$\frac{d}{dt}\left(\hat{d}_{s_i/c}(t)\right) = \begin{cases} \eta_{s_i,1}(t) + k_\xi\Xi_{s_i}(t)\tilde{d}_{s_i/c}(t), & t < \tau_{s_i} \\ \eta_{s_i,1}(t) + \left(k_1 + k_\xi\Xi_{s_i}(t)\right)\tilde{d}_{s_i/c}(t), & t \geq \tau_{s_i} \end{cases} \tag{24}$$

TABLE I
RMS Depth Error and Position Error in Meters Over 15 Experiments

Experiment	Estimator 1	Estimator 2	Estimator 3	Estimator 4	Trajectory
1	70.499	55.525	58.325	55.435	0.017
2	66.358	42.065	52.827	42.838	0.026
3	80.466	64.701	67.616	63.706	0.037
4	76.250	59.583	65.593	59.565	0.032
5	79.419	63.244	70.798	65.674	0.032
6	82.611	69.300	71.996	68.558	0.018
7	65.971	48.192	57.287	46.392	0.020
8	73.864	60.002	64.290	59.858	0.027
9	77.699	61.468	69.227	60.708	0.020
10	72.053	55.916	63.679	55.771	0.023
11	75.472	59.757	59.916	60.798	0.016
12	77.571	63.623	66.451	63.686	0.022
13	83.663	67.817	72.995	68.274	0.036
14	74.030	58.764	66.941	59.245	0.030
15	77.742	56.067	64.983	56.612	0.029
Mean	75.578	59.068	64.862	59.141	0.026
Standard Deviation	5.086	6.819	5.522	6.938	0.007

TABLE II
RMS Depth Error and Position Error in Meters Over 15 Experiments Before Learning

Experiment	Estimator 1	Estimator 2	Estimator 3	Estimator 4	Trajectory
1	133.393	105.884	109.492	104.984	0.007
2	137.855	87.811	107.410	88.252	0.005
3	137.133	110.694	114.093	108.139	0.005
4	134.047	105.321	113.742	104.732	0.005
5	138.285	110.761	120.448	112.189	0.005
6	137.122	115.571	118.032	113.951	0.005
7	126.254	92.370	103.992	87.503	0.004
8	128.662	104.859	109.866	103.421	0.004
9	136.862	108.817	119.151	106.569	0.005
10	128.454	100.033	110.136	98.836	0.004
11	137.677	109.201	107.660	110.506	0.006
12	132.034	108.494	111.989	108.105	0.005
13	139.927	113.816	119.680	112.477	0.009
14	131.922	104.987	114.095	101.154	0.005
15	142.222	103.213	116.301	103.489	0.006
Mean	134.790	105.455	113.073	104.287	0.005
Standard Deviation	4.440	7.212	4.825	7.645	0.001

where $\Xi_{s_i}(t) \triangleq \xi_{s_i}^T(t)\xi_{s_i}(t)$. Substituting (24) into the time derivative of (13) yields

$$\frac{d}{dt}\left(\tilde{d}_{s_i/c}(t)\right) = \begin{cases} -k_\xi\Xi_{s_i}(t)\tilde{d}_{s_i/c}(t), & t < \tau_{s_i} \\ -\left(k_1 + k_\xi\Xi_{s_i}(t)\right)\tilde{d}_{s_i/c}(t), & t \geq \tau_{s_i}. \end{cases} \tag{25}$$

Remark 2: Under Assumption 5, $\Xi_{s_i}(t) \geq 0$ given $\|\underline{v}_c(t)\|$ may be zero for any period of time; however, for Assumption 5 to be satisfied, there will be times where $\Xi_{s_i}(t) > 0$. Specifically, there will exist a set of times $\mathcal{T}_{s_i} \subset \bigcup_{h_i=1}^{N}(t_{h_i} - \varsigma, t_{h_i})$ such that $\Xi_{s_i}(t) > 0 \quad \forall t \in \mathcal{T}_{s_i}$, where h_i and t_{h_i} are from (9), implying the design in (23) may improve transient performance under Assumption 5. The reasoning is that for Assumption 5 to hold, there must be a change in $\underline{u}_{s_i/c}(t)$ over the time intervals $\mathcal{T}_{s_i} \subset \bigcup_{h_i=1}^{N}(t_{h_i} - \varsigma, t_{h_i})$ implying $\|\xi_{s_i}(t)\| > 0$ since $\xi_{s_i}(t) \triangleq (\frac{d}{dt}(\underline{u}_{s_i/c}(t)) + \underline{\omega}_c^{\times}(t)\underline{u}_{s_i/c}(t))$. Since $\Xi_{s_i}(t) \triangleq \xi_{s_i}^T(t)\xi_{s_i}(t)$, $\Xi_{s_i}(t) > 0$ over those time intervals implying $\frac{d}{dt}(\tilde{d}_{s_i/c}(t)) = -k_\xi\Xi_{s_i}(t)\tilde{d}_{s_i/c}(t) < 0$, and hence, $\tilde{d}_{s_i/c}(t)$ will be decaying.

Remark 3: As shown in (10)–(12), the distance values are determined and may be used directly; however, Assumption 5 describes the period of time τ_{s_i} required to learn (10) where no information about the distances is available except through the dynamics. Additionally, feature tracking is noisy resulting in noisy estimates of the rotation $R_{k/c}(t)$ and unit vector $\underline{u}_{k/c}(t)$ implying the history stack will have noisy data. The estimators in (16)–(18) combine the dynamics with feedback resulting in improved estimates and robustness to noise. Specifically, during the learning period, $t < \tau_{s_i}$, the estimators ensure the error is bounded as subsequently shown in the analysis. After the learning period, $t \geq \tau_{s_i}$, the estimators use error feedback to converge to the value by effectively filtering the measurements [e.g., from (16), $\frac{d}{dt}(\hat{d}_{s_i/c}(t)) \triangleq \eta_{s_i,1}(t) + k_1(\nu_{s_i,1}(t) - \hat{d}_{s_i/c}(t))$ after $t \geq \tau_{s_i}$ implying a control over how much a noisy $\nu_{s_i,1}(t)$ can affect the state estimate through k_1). Additionally, using the extended estimator in (23) further improves transient performance as subsequently shown in Fig. 4 where the estimator has practically converged when $t = \max\{\tau_{s_i}\}$, the time all the distances have been learned. Furthermore, the results in Tables I – III show that (23) outperforms the other methods

TABLE III
RMS DEPTH ERROR AND POSITION ERROR IN METERS OVER 15 EXPERIMENTS AFTER LEARNING

Experiment	Estimator 1	Estimator 2	Estimator 3	Estimator 4	Trajectory
1	13.374	3.979	12.838	8.567	0.017
2	10.655	1.868	13.894	8.166	0.029
3	12.513	2.657	14.246	10.123	0.045
4	12.517	1.180	15.656	7.629	0.038
5	13.431	1.903	20.777	17.759	0.038
6	13.359	2.414	16.889	7.562	0.022
7	10.403	3.502	22.177	10.390	0.024
8	11.632	3.804	17.099	11.610	0.033
9	12.661	1.683	19.959	9.847	0.024
10	11.162	1.861	20.035	9.457	0.028
11	10.239	2.828	13.483	8.119	0.019
12	9.996	1.576	12.845	7.833	0.027
13	13.093	3.261	19.967	16.608	0.044
14	11.044	3.090	24.677	21.283	0.036
15	13.885	2.599	18.542	8.494	0.034
Mean	11.998	2.547	17.539	10.897	0.031
Standard Deviation	1.284	0.831	3.567	4.080	0.008

in terms of rms error. An approach that only uses (10)–(12) would imply no information is available until enough data are collected, which will have higher rms error and no control over the rate of learning.

V. STABILITY ANALYSIS

Given the observer in (23) is an extension of (16), the resulting stability analysis of (16) is identical to Theorem 1 and is excluded. Let $\eta(t) \triangleq [\tilde{d}_{s_i/c}(t) \quad \tilde{d}_{k/c}(t) \quad \tilde{d}_{s_i/k}(t)]^T$ and $V(\eta(t)) : \mathbb{R}^3 \to \mathbb{R}$ be a candidate Lyapunov function defined as

$$V(\eta(t)) \triangleq \frac{1}{2}\eta^T(t)\eta(t). \tag{26}$$

Theorem 1: The observer update laws defined in (17), (18), and (23) ensure the estimation errors in $\eta(t)$ are bounded and globally exponentially stable in the sense that

$$\|\eta(t)\| \leq \|\eta(0)\| \exp(\beta \tau_{s_i}) \exp(-\beta t). \tag{27}$$

Proof: Taking the time derivative of (26), then substituting the error derivatives in (20), (21), and (25), simplifying, then upper bounding yields

$$\frac{d}{dt}(V(\eta(t))) \leq \begin{cases} 0, & t < \tau_{s_i} \\ -2\beta V(\eta(t)), & t \geq \tau_{s_i} \end{cases} \tag{28}$$

where $\beta = \min\{k_1, k_2, k_3\}$. From (26) and (28), [48, Th. 8.4] can be invoked to conclude that $\|\eta(t)\|^2 \leq \|\eta(0)\|^2 \quad \forall t \leq \tau_{s_i}$. From [48, Th. 4.10], $\|\eta(t)\|^2 \leq \|\eta(\tau_{s_i})\|^2 \exp(2\beta\tau_{s_i}) \exp(-2\beta t) \quad \forall t \geq \tau_{s_i}$. Evaluating the first bound on $\|\eta(t)\|^2$ at $t = \tau_{s_i}$, then substituting into the second bound on $\|\eta(t)\|^2$ and taking the square root yields (27). ■

VI. EXPERIMENTAL RESULTS

Fifteen experiments are provided to demonstrate the performance of the developed observers. The performances of the developed observers in (16)–(18) and (23) were tested using the Eigen3, OpenCV, and ROS c++ libraries (cf., [49], [50], and [51], respectively). A Kobuki Turtlebot with a 1920×1080 monochrome iDS uEye camera, shown in Fig. 2, provides images at 30 Hz. Features were extracted from images of a checkerboard, shown in Fig. 2, with 8×6 corners (48 total features) where each square is 0.06 m $\times$ 0.06 m in dimension. The linear and angular velocities of the camera were calculated using the Turtlebot wheel encoders and a gyroscope at 50 Hz. In addition, an Optitrack motion capture system operating at 120 Hz measured the pose of the camera and checkerboard, allowing for the position of each feature relative to the camera to be known for comparison. Image processing and estimators were multithreaded using four threads and executed simultaneously on a computer with an Intel i7 processor running at 3.4 GHz, ensuring the system ran at 30 Hz. The errors of the distance estimators in (16) and (23) are compared to the estimator in [21] and an EKF. Given the estimator in [21] and the EKF estimate the inverse depth (i.e., $\frac{1}{z_{s_i/c}(t)}$, where $z_{s_i/c}(t)$ is the depth to feature s_i from c expressed in $\mathcal{F}_c$), whereas the estimators in (16) and (23) estimate the distance, the comparison of the four methods is shown by the depth, $z_{s_i/c}(t)$, which is the third element of $u_{s_i/c}(t)d_{s_i/c}$.

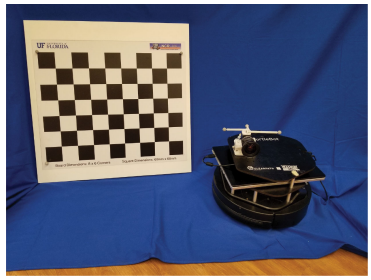

Fig. 2. Image shows the checkerboard, Kobuki Turtlebot, and iDS uEye camera used for experiments.

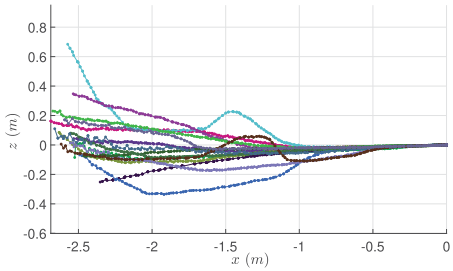

Fig. 3. Camera trajectories for each of the 15 experiments.

For each experiment, the Turtlebot started approximately 3 m away from the checkerboard, and various trajectories were taken, shown in Fig. 3, while maintaining the checkerboard in the FOV. In each experiment, the Turtlebot initially started at rest, and after traveling 2.5 m, the estimators were stopped to provide a large baseline. After the Turtlebot started its motion in the beginning of each experiment, the Turtlebot traveled without stopping until after the estimators were stopped in an effort to have the ideal conditions for estimation (i.e., continuous motion of the features in the camera FOV and continuous linear motion of the camera as is required for [21] and the EKF). The initial distances for the estimators in (16), (18), (23), the estimator in [21] and the EKF were initialized from a depth of 0.5 m. The estimator in (17) was initialized to 0 m. The gains for (16)–(18) and (23) were selected as $k_1 = k_2 = k_3 = 25$ and $k_\xi = 25k_1$, respectively. The maximum value for ς was 5 s. The 48 feature estimates were combined using a mean at each instance to update (17). The gain for the method in [21] was selected to be 100. The covariance matrices for the EKF were selected as $R = r\left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$ for the measurement covariance, $Q = r\begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100000 \end{bmatrix}$ for the process covariance, and $P(0) = r\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 150000 \end{bmatrix}$ for the initial state covariance, where $r = 0.00001$. The gains and covariance matrices used were experimentally determined to obtain the performance shown in Figs. 3–5 and Tables I–III.

Remark 4: For a general system, the optimal approach to select good data and remove bad data for (9) (e.g., due to noise or parameter changes) remains an open problem and is often left to intuition about the system. Results in [47] demonstrated a purging method to remove bad data using two separate history stacks. The first stack was actively

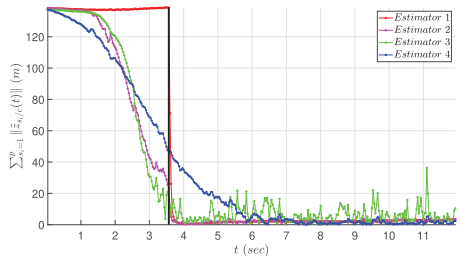

Fig. 4. Sum of the norm of each depth error across the 48 features (i.e., $\sum_{s_i=1}^{48} \|\tilde{z}_{s_i/c}(t)\|$) for experiment 11. Estimator 1 (red) refers to (16), estimator 2 (magenta) refers to (23), estimator 3 (green) refers to Dani *et al.* [21], and estimator 4 (blue) refers to the EKF. The black vertical line indicates the time when enough information was collected for learning.

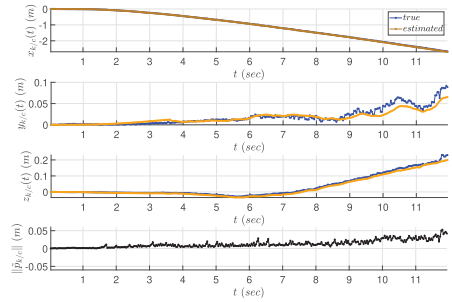

Fig. 5. Position error of the camera and the distance error using the estimator in (17) for experiment 11.

used by the estimator while the second stack collected data using a singular value maximization method. After a set of conditions was satisfied, the first stack was replaced by the second stack and the second stack was purged. After purging the second stack, new data were again collected using the singular value maximization method and the process repeated. For the experiments in this article, the selection of data was based on knowledge of approximate noise magnitudes in feature tracking and velocity measurements. Specifically, data were only selected if $\|\mathcal{Y}_{s_i}(t)\| \geq \epsilon_{\mathcal{Y}}$ and $\|\mathcal{U}_{s_i}(t)\| \geq \epsilon_{\mathcal{U}}$ where $\epsilon_{\mathcal{Y}}, \epsilon_{\mathcal{U}} \in \mathbb{R}_{>0}$ are values selected based on an empirical process. Because $\mathcal{Y}_{s_i}(t)$ is full column rank when $\|\mathcal{Y}_{s_i}(t)\| \geq \epsilon_{\mathcal{Y}}$ and $\|\mathcal{U}_{s_i}(t)\| \geq \epsilon_{\mathcal{U}}$, the value of $d_{s_i/k}$ approximated by $\mathcal{Y}_{s_i}(t)$ and $\mathcal{U}_{s_i}(t)$ can be determined. Given $d_{s_i/k} > 0$, and some knowledge about reasonable values for the distances, values of $d_{s_i/k}$ can be determined and only $\mathcal{Y}_{s_i}(t)$ and $\mathcal{U}_{s_i}(t)$ values that had $d_{s_i/k}$ estimates falling in these bounds are saved to $\Sigma_{\mathcal{Y}_{s_i}}$ and $\Sigma_{\mathcal{U}_{s_i}}$. The values for $\epsilon_{\mathcal{Y}}$ and $\epsilon_{\mathcal{U}}$ were $\epsilon_{\mathcal{Y}} = \epsilon_{\mathcal{U}} = 0.1$ and the bounds on the distance were selected as 0.5 and 6 m.

A comparison of the example performance over time of the estimators is shown in Fig. 4 and Tables I–III, where before learning refers to $t < \max\{\tau_{s_i}\}$ and after learning refers to $t \geq \max\{\tau_{s_i}\}$. Fig. 4 shows the typical learning time where $\max\{\tau_{s_i}\} = 3.6$ s is shown as a black vertical line. Fig. 4 shows a comparison of the sum of the norm of each depth error across the 48 features (i.e., $\sum_{s_i=1}^{48} \|\tilde{z}_{s_i/c}(t)\|$) on the checkerboard for the estimators in (16), (23), [21], and the EKF, respectively. As shown in Fig. 4, the EKF estimator starts converging the fastest, but reaches steady state slower than the estimators in (16), (23), and [21]. However, after converging, the EKF has a similar error to the estimators in (16) and (23). Fig. 4 also shows that the estimator in (16) does not converge until sufficient learning occurs (at $t = 3.6$ s for experiment 11). The extension in Section IV shows an advantage of using current input–output data in the estimator, as shown by the mean rms errors in Tables I and II. Specifically, the estimator in (16) is at a disadvantage to the other estimators before sufficient excitation has occurred, whereas the estimator in (23) starts converging to the true depths at a similar time frame as the estimator in [21]. The average rms error of (16) is more than 10 m greater than the other estimators over the entirety of each experiment, and more than 20 m greater before learning. However, Table III shows that the average error of (16) after learning is only 1 m greater than the EKF on average.

The extension in Section IV, specifically the design in (23), improves the error convergence of (16) such that the rms error is lower than the EKF on average. As shown in Table I, the average error over the entire experiment runtime was 59.068 m for (23) compared to 59.141 m for the EKF. After learning, the average rms error for the estimator in (23) was smaller (2.547 m) compared to the EKF (10.897 m). However, as shown in Table II, the rms error before learning was smaller for the EKF compared to (23), where the errors were 104.287 m for the EKF and 105.455 m for the estimator in (23). Additionally, Tables I–III show that the design in (23) has a smaller rms error than the design in [21] on average. Fig. 5 and Tables I–III show that the position error using (17) is small with an average rms error of 0.026 m over the entire run; 0.005 m before learning and 0.031 m after learning, which is approximately 1.2% error relative to trajectory length. The error increase after learning is a result of noise, which, as shown in Fig. 4 and Table III, causes the depth error to remain small but bounded at approximately 1.8% of the initial error. These experimental results demonstrate the ability of the observer in (16) to leverage both immediate information and learning to both converge quickly with low rms error and maintain a low rms error after converging.

VII. Conclusion

Novel observers using a single camera and structure from motion theory are developed to estimate the Euclidean distance to features on a stationary object and the Euclidean trajectory the camera takes while observing the object. Unlike previous results that estimate the inverse depth to features, the developed observer for estimating the Euclidean distance to features does not require the positive depth constraint. A Lyapunov-based stability analysis shows that the observer error is exponentially converging without requiring PE and instead only requires finite excitation through the use of ICL. An experimental comparison of the developed estimator to existing estimators shows that it achieves lower rms error when comparing feature depth estimates on average and the rms error of the position also remains low.

Future work may examine extending this result to include a solution for estimating the velocities of the camera and the structure of multiple objects while allowing those objects to leave the camera FOV over time either temporarily or permanently. The extended result would allow a camera to travel over larger distances and allow the camera to reconstruct a larger environment, which may be used in the development of a SLAM algorithm.

VII. Acknowledgment

Any opinions, findings, and conclusions or recommendations expressed in this article are those of the author(s) and do not necessarily reflect the views of the sponsoring agency.

References

[1] G. Dubbelman and B. Browning, "COP-SLAM: Closed-form online pose-chain optimization for visual slam," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1194–1213, Oct. 2015.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[3] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[4] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: A survey from 2010 to 2016," *IPSJ Trans. Comput. Vis. Appl.*, vol. 9, no. 1, 2017, Art. no. 16.

[5] M. Karrer, P. Schmuck, and M. Chli, "CVI-SLAM—Collaborative visual-inertial SLAM," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 2762–2769, Oct. 2018.

[6] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[7] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3-D Vision*. Vienna, Austria: Springer, 2004.

[8] L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithm for estimating depth from image sequences," *Int. J. Comput. Vis.*, vol. 3, pp. 209–236, 1989.

[9] M. Jankovic and B. Ghosh, "Visually guided ranging from observations points, lines and curves via an identifier based nonlinear observer," *Syst. Control Lett.*, vol. 25, no. 1, pp. 63–73, 1995.

[10] S. Soatto, R. Frezza, and P. Perona, "Motion estimation via dynamic vision," *IEEE Trans. Autom. Control*, vol. 41, no. 3, pp. 393–413, Mar. 1996.

[11] H. Kano, B. K. Ghosh, and H. Kanai, "Single camera based motion and shape estimation using extended Kalman filtering," *Math. Comput. Model.*, vol. 34, pp. 511–525, 2001.

[12] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "Structure from motion causally integrated over time," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 523–535, Apr. 2002.

[13] W. E. Dixon, Y. Fang, D. M. Dawson, and T. J. Flynn, "Range identification for perspective vision systems," *IEEE Trans. Autom. Control*, vol. 48, no. 12, pp. 2232–2238, Dec. 2003.

[14] X. Chen and H. Kano, "State observer for a class of nonlinear systems and its application to machine vision," *IEEE Trans. Autom. Control*, vol. 49, no. 11, pp. 2085–2091, Nov. 2004.

[15] D. Karagiannis and A. Astolfi, "A new solution to the problem of range identification in perspective vision systems," *IEEE Trans. Autom. Control*, vol. 50, no. 12, pp. 2074–2077, Dec. 2005.

[16] D. Braganza, D. M. Dawson, and T. Hughes, "Euclidean position estimation of static features using a moving camera with known velocities," in *Proc. IEEE Conf. Decis. Control*, New Orleans, LA, USA, Dec. 2007, pp. 2695–2700.

[17] A. De Luca, G. Oriolo, and P. R. Giordano, "Feature depth observation for image-based visual servoing: Theory and experiments," *Int. J. Robot. Res.*, vol. 27, no. 10, pp. 1093–1116, 2008.

[18] G. Hu, D. Aiken, S. Gupta, and W. Dixon, "Lyapunov-based range identification for a paracatadioptric system," *IEEE Trans. Autom. Control*, vol. 53, no. 7, pp. 1775–1781, Aug. 2008.

[19] F. Morbidi and D. Prattichizzo, "Range estimation from a moving camera: An immersion and invariance approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, May 2009, pp. 2810–2815.

[20] N. Zarrouati, E. Aldea, and P. Rouchon, "SO(3)-invariant asymptotic observers for dense depth field estimation based on visual data and known camera motion," in *Proc. Amer. Control Conf.*, Montreal, QC, Canada, Jun. 2012, pp. 4116–4123.

[21] A. Dani, N. Fischer, Z. Kan, and W. E. Dixon, "Globally exponentially stable observer for vision-based range estimation," *Mechatronics*, vol. 22, no. 4, pp. 381–389, 2012.

[22] A. Dani, N. Fischer, and W. E. Dixon, "Single camera structure and motion," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 241–246, Jan. 2012.

[23] Z. I. Bell, H.-Y. Chen, A. Parikh, and W. E. Dixon, "Single scene and path reconstruction with a monocular camera using integral concurrent learning," in *Proc. IEEE Conf. Decis. Control*, 2017, pp. 3670–3675.

[24] A. Parikh, R. Kamalapurkar, H.-Y. Chen, and W. E. Dixon, "Homography based visual servo control with scene reconstruction," in *Proc. IEEE Conf. Decis. Control*, 2015, pp. 6972–6977.

[25] J. Oliensis, "A critique of structure-from-motion algorithms," *Comput. Vis. Image Underst.*, vol. 80, pp. 172–214, 2000.

[26] J. Oliensis and R. Hartley, "Iterative extensions of the Strum/Triggs algorithm: Convergence and nonconvergence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2217–2233, Dec. 2007.

[27] F. Kahl and R. Hartley, "Multiple-view geometry under the L_∞-norm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1603–1617, Sep. 2008.

[28] A. Parikh, T.-H. Cheng, H.-Y. Chen, and W. E. Dixon, "A switched systems framework for guaranteed convergence of image-based observers with intermittent measurements," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 266–280, Apr. 2017.

[29] A. Parikh, T.-H. Cheng, R. Licitra, and W. E. Dixon, "A switched systems approach to image-based localization of targets that temporarily leave the camera field of view," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 6, pp. 2149–2156, Nov. 2018.

[30] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Target tracking in the presence of intermittent measurements via motion model learning," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 805–819, Jun. 2018.

[31] M. Boutayeb, H. Rafaralahy, and M. Darouach, "Convergence analysis of the extended Kalman filter used as an observer for nonlinear deterministic discrete-time systems," *IEEE Trans. Autom. Control*, vol. 42, no. 4, pp. 581–586, Apr. 1997.

[32] K. Reif and R. Unbehauen, "The extended Kalman filter as an exponential observer for nonlinear systems," *IEEE Trans. Signal Process.*, vol. 47, no. 8, pp. 2324–2328, Aug. 1999.

[33] G. V. Chowdhary and E. N. Johnson, "Theory and flight-test validation of a concurrent-learning adaptive controller," *J. Guid. Control Dyn.*, vol. 34, no. 2, pp. 592–607, Mar. 2011.

[34] G. Chowdhary, M. Mühlegg, J. How, and F. Holzapfel, "Concurrent learning adaptive model predictive control," in *Advances in Aerospace Guidance, Navigation and Control*, Q. Chu, B. Mulder, D. Choukroun, E.-J. van Kampen, C. de Visser, and G. Looye, Eds. Berlin, Germany: Springer, 2013, pp. 29–47.

[35] G. Chowdhary, T. Yucelen, M. Mühlegg, and E. N. Johnson, "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *Int. J. Adapt. Control Signal Process.*, vol. 27, no. 4, pp. 280–301, 2013.

[36] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for approximate optimal regulation," *Automatica*, vol. 64, pp. 94–104, 2016.

[37] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Integral concurrent learning: Adaptive control with parameter convergence using finite excitation," *Int. J. Adapt. Control Signal Process.*, vol. 33, no. 12, pp. 1775–1787, Dec. 2019.

[38] Z. Bell, P. Deptula, H.-Y. Chen, E. Doucette, and W. E. Dixon, "Velocity and path reconstruction of a moving object using a moving camera," in *Proc. Amer. Control Conf.*, 2018, pp. 5256–5261.

[39] Z. Bell, J. Nezvadovitz, A. Parikh, E. Schwartz, and W. Dixon, "Global exponential tracking control for an autonomous surface vessel: An integral concurrent learning approach," *IEEE J. Ocean Eng.*, vol. 45, no. 2, pp. 362–370, Apr. 2020.

[40] H.-Y. Chen, Z. I. Bell, P. Deptula, and W. E. Dixon, "A switched systems framework for path following with intermittent state feedback," *IEEE Control Syst. Lett.*, vol. 2, no. 4, pp. 749–754, Oct. 2018.

[41] H.-Y. Chen, Z. Bell, P. Deptula, and W. E. Dixon, "A switched systems approach to path following with intermittent state feedback," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 725–733, Jun. 2019.

[42] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 593–600.

[43] J.-Y. Bouguet, "Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, no. 1–10, p. 4, 2001.

[44] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp. 674–679.

[45] K. Fathian, J. P. Ramirez-Paredes, E. A. Doucette, J. W. Curtis, and N. R. Gans, "QuEst: A quaternion-based approach for camera motion estimation from minimal feature points," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 857–864, Apr. 2018.

[46] G. Chowdhary and E. Johnson, "A singular value maximizing data recording algorithm for concurrent learning," in *Proc. Amer. Control Conf.*, 2011, pp. 3547–3552.

[47] R. Kamalapurkar, B. Reish, G. Chowdhary, and W. E. Dixon, "Concurrent learning for parameter estimation using dynamic state-derivative estimators," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3594–3601, Jul. 2017.

[48] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[49] G. Guennebaud *et al.*, "Eigen v3," 2010. [Online]. Available: http://eigen.tuxfamily.org

[50] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.

[51] M. Quigley *et al.*, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, Kobe, Japan, 2009, vol. 3, no. 3.2, p. 5.