Target Tracking in the Presence of Intermittent Measurements via Motion Model Learning

Anup Parikh[1], Rushikesh Kamalapurkar[2] and Warren E. Dixon[1]

***Abstract*—When using a camera to estimate the pose of a moving target, measurements may only be available intermittently, due to feature tracking losses from occlusions or the limited field of view of the camera. Results spanning back to the Kalman filter have demonstrated the utility of using a predictor to update state estimates when measurements are not available, but target velocity measurements or a motion model must be known to implement a predictor for image-based pose estimation. In this paper, a novel estimator and predictor are developed to simultaneously learn a motion model, and estimate the pose, of a moving target from a moving camera. A stability analysis is provided to prove convergence of the state estimates and function approximation without requiring the restrictive persistent excitation condition. Two experiments illustrate the performance of the developed estimator and predictor. One experiment involves a stationary camera observing a mobile robot with sporadic feature tracking losses, and a second experiment involves a quadcopter moving between two mobile robots on a road network.**

***Index Terms*—Target Tracking, Switched Systems, Estimation, Adaptive Methods**

I. Introduction

A number of advances in imaging and computer vision have enabled geometric reconstruction of features through image feedback. Due to the projection in the imaging process, and the resulting scale ambiguity, typical approaches exploit multiple views of the scene, as well as scale information, to recover the Euclidean geometry, e.g., stereo vision with a known baseline or structure from motion (SfM) with known camera motion. For online reconstruction (i.e., recursive methods) using a single camera, a number of observer/filtering techniques have been developed to solve the SfM problem (i.e., determine the relative Euclidean coordinates of an object with respect to a camera) for a stationary object and moving camera (cf. [1]–[7]), a moving object with a stationary camera (cf. [7]–[12]), as well as for the case where both the object and camera are in motion (cf. [13], [14]). In all cases, velocity information of either the camera or target, or both, is used to inject scale information into the system.

[1]Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville FL 32611-6250, USA Email:{anuppari, wdixon}@ufl.edu
[2]School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater OK 74078, USA Email:rushikesh.kamalapurkar@okstate.edu

This research is supported in part by a Task Order contract with the Air Force Research Laboratory, Munitions Directorate at Eglin AFB, US National Geospatial Intelligence Agency Grant HM0177-12-1-0006 and US Department of Agriculture National Robotics Initiative Grant 2013-67021-21074. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsoring agency.

Previously developed SfM observers rely on continuous measurement availability to show convergence of the estimates. However, in many applications, image feature measurements may be intermittently unavailable due to feature tracking losses, occlusions, limited camera field of view (FOV), or even the finite frame rate of the camera, which result in intermittent measurements. In this paper, a novel estimator and predictor are developed and shown to converge to within an error bound despite the intermittent measurements. Due to the different modes of operation (i.e., when measurements are available versus when measurements are unavailable), the estimation error obeys different dynamics in each mode, and switched systems theory is used to analyze the overall stability and performance of the system. Switched systems methods are necessary due to the well known result that switching between stable systems can lead to instability [15]. The problem is exacerbated in this paper, as shown in the analysis, because the estimation error dynamics are unstable when measurements are unavailable. Therefore, additional analysis is necessary to demonstrate that, despite intermittent measurements, the overall switched estimation error dynamics are stable.

Numerous results have been developed for feature tracking in the presence of intermittent visibility of the target. For example, [16] and [17] describe methods for learning a motion model online for feature motion prediction. Similarly, in [18], [19] Kalman or particle filters are used to estimate feature motion and predict feature coordinates while occluded. In contrast, [20]–[22] use visual context to increase the robustness of feature trackers to occlusions. For the SfM problem, a technique that is robust to occlusions or feature tracking losses is developed in [23]; however, only the shape of the object is recovered, and not the 3D position due to the orthogonal projection model used. In contrast to such results, the full 6 degree of freedom (DOF) pose of the target is estimated in this paper, and the estimates are shown to be stable.

Many of the probabilistic approaches for SfM, or the associated simultaneous localization and mapping (SLAM) problem, utilize a predictor similar to that developed in this paper or circumvent the intermittent sensing issue by only updating state estimates when new measurements are available (see [24] and [25] for an overview). However, these approaches are based on either linearizations of the nonlinear dynamics (cf. [26]–[31]), and therefore only show local convergence, or are sample based (e.g., [32] and [33]), and therefore can only show optimal estimation in the limit as the number of samples approach infinity. Much of the recent literature on target tracking has focused on using suboptimal algorithms for tracking using simplified motion models (e.g., constant

velocity, constant turn rate, etc.), with a focus on reduced complexity and improving practical performance, and do not analyze estimation error growth due to model uncertainty or show estimation error convergence [34], [35]. Some methods explicitly handle occlusions, though they either assume availability of range measurements and only estimate position, therefore rendering the system linear (cf. [36]–[38]), or only estimate relative depth ordering and do not consider the pose estimation problem, e.g., [39]. Other methods learn a model of the target motion online using function approximation methods (cf. [40]–[47]), though do not provide a convergence analysis. Conversely, the full nonlinear dynamics are analyzed in this paper, resulting in an arbitrarily large region of attraction around the zero estimation error trajectory (put more simply, the estimator converges for any set of initial conditions, rather than the linearization based approaches that rely on a good initial guess for practical performance to match their theoretical local convergence results), and the proposed estimator-predictor structure has computing requirements that can be met by typical or low-end modern computers (see Section VII). Furthermore, convergence and consistency proofs of probabilistic estimators typically require knowledge of the probability distribution of the uncertainty in the system, and result in convergence in mean or in mean square. In comparison, analysis of deterministic observers typically assume boundedness and some level of smoothness of disturbances, and yield asymptotic or exponential convergence. The primary contribution of this paper is in the development and analysis of a novel estimator and predictor that ensures convergence to an ultimate bound as well as online learning of a motion model of the target using a deterministic framework.

Our previous results have shown stability of the position estimation error during intermittent measurements [48], [49], provided dwell time conditions are satisfied. In switched systems, dwell time conditions specify the minimum amount of time a single mode or system must be active before switching to maintain system stability, and reverse dwell time conditions specify the maximum amount of time a system can remain active to maintain system stability. These conditions must be met at every switch from one system to another. In the context of target tracking with intermittent measurements, the dwell time conditions specify the minimum contiguous duration the target must remain in view, and reverse dwell time conditions specify the maximum contiguous duration the target can remain out of view. In [49], a zero-order hold is performed on the state estimates when measurements are unavailable, resulting in growth of the estimation error based on the trigonometric tangent function, and an ultimately bounded estimation error result. Since the tangent function is unbounded for finite arguments, (reverse) dwell time conditions are necessary at every period in which measurements are (un)available to ensure stability. In [48], a predictor was used to update the state estimates when measurements are unavailable. This results in exponential growth of the estimation errors when measurements are unavailable, allowing the use of average dwell times for stability. Average dwell time conditions are easier to satisfy than regular dwell time conditions since they only restrict the average mode durations rather than every duration (e.g., if a target is in view for 3 seconds, then out of view, and then in view again for 1 second, an average dwell time condition of 2 seconds is satisfied, but a dwell time condition of 2 seconds is not). One downside of the approach in [48] is that the use of a predictor requires knowledge of a motion model of the target to generate target velocity signals utilized in the predictor.

In this paper, the target motion model is learned online. A number of adaptive methods have been developed to compensate for unknown functions or parameters in the dynamics; however, parameter estimates may not approach the true parameters without persistent excitation (PE) [50]–[52]. The PE condition cannot be guaranteed *a priori* for nonlinear systems (as apposed to linear systems, e.g., [50, Theorem 5.2.1]), and is difficult to check online, in general. Recently, a technique known as concurrent learning (CL) was developed to use recorded data for online parameter estimation [53]–[55] with an alternative excitation condition. In CL, input and state derivatives are recorded and used similar to recursive least squares to establish a negative definite parameter estimation error term in the Lyapunov analysis, and hence a negative definite Lyapunov derivative provided a *finite excitation* condition is satisfied. However, state derivatives can be noisy, and require extensive filter design and tuning to yield satisfactory signals for use in CL. A further contribution of this paper is that the CL technique is reformulated in terms of an integral (ICL), removing the need for state derivatives, while preserving convergence guarantees. Compared to traditional adaptive methods that utilize PE to ensure parameter convergence, and hence exponential stability, ICL only requires excitation for a finite period of time, and the excitation condition can be checked online.

In this paper, data is recorded online when measurements are available (i.e., the target is in view of the camera). Using the ICL technique, a motion model of the target is learned, and used in a predictor to estimate target pose when it is not visible to the camera. A stability analysis is provided to show that this estimation and prediction with learning scheme yields an estimate of the target pose that converges to within an arbitrarily small bound around the true target pose.

Two experiments are included to illustrate the performance of this estimator. One experiment involves a stationary camera observing a mobile robot moving according to static vector field. The results of this experiment demonstrate that the learning component of the estimator quickly converges and accurately predicts the robot motion. The second experiment involves a moving camera observing two mobile robots moving along a road network. Due to the limited FOV and resolution of the camera, measurements of both robots are not available simultaneously. Despite the intermittent measurements, and the stochastic motions of the robots at various points on the road network, the results demonstrate the feasibility of the developed approach for tracking the pose of multiple targets.

II. System Dynamics

Figure 1 is used to develop the image kinematics. In Figure 1, $\mathcal{F}_G$ denotes a fixed inertial reference frame with an

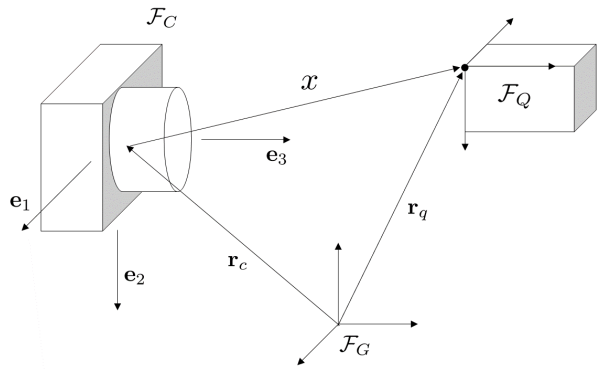

Figure 1. Reference frames and coordinate systems of a moving camera observing a moving target.

arbitrarily selected origin and Euclidean coordinate system, $\mathcal{F}_Q$ denotes a reference frame fixed to the moving object, with an arbitrarily selected origin and Euclidean coordinate system, and $\mathcal{F}_C$ denotes a reference frame fixed to the camera. The right handed coordinate system attached to $\mathcal{F}_C$ has its origin at the principal point of the camera, $e_3 \in \mathbb{E}^3$ axis pointing out and collinear with the optical axis of the camera, $e_1 \in \mathbb{E}^3$ axis aligned with the horizontal axis of the camera, and $e_2 \triangleq e_3 \times e_1 \in \mathbb{E}^3$. The vectors $r_q \in \mathbb{E}^3$ and $r_c \in \mathbb{E}^3$ represent the vectors from the origin of $\mathcal{F}_G$ to the origins of $\mathcal{F}_Q$ and $\mathcal{F}_C$, respectively. The kinematics of the coordinates of the relative position vector expressed in the camera coordinate system are

$$\dot{x} = v_q - v_c - \omega_c^{\times} x \tag{1}$$

where $x \in \mathbb{R}^3$ denotes the position of the origin of $\mathcal{F}_Q$ with respect to the origin of $\mathcal{F}_C$ (i.e., the relative position of the object with respect to the camera), $v_q \triangleq \begin{bmatrix} v_{q1} & v_{q2} & v_{q3} \end{bmatrix}^T \in \mathbb{R}^3$ is the velocity of the origin of $\mathcal{F}_Q$ with respect to the origin of $\mathcal{F}_G$ (i.e., the inertial linear velocity of the object), $v_c \triangleq \begin{bmatrix} v_{c1} & v_{c2} & v_{c3} \end{bmatrix}^T \in \mathbb{R}^3$ is the linear velocity of the origin of $\mathcal{F}_C$ with respect to the origin of $\mathcal{F}_G$ (i.e., the inertial linear velocity of the camera), and $\omega_c \triangleq \begin{bmatrix} \omega_{c1} & \omega_{c2} & \omega_{c3} \end{bmatrix}^T \in \mathbb{R}^3$ is the angular velocity of $\mathcal{F}_C$ with respect to $\mathcal{F}_G$ (i.e., the inertial angular velocity of the camera), all expressed in the camera coordinate system. Also, $()^{\times} : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ represents the skew operator, defined as

$$p^{\times} \triangleq \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix}.$$

In the following analysis, the quaternion parameterization will be used to represent orientation. Let $\underline{q} \in \mathbb{H}$ be the unit quaternion parameterization of the orientation of the object with respect to the camera, which can be represented in the four dimensional vector space $\mathbb{R}^4$ using the standard basis 1, i, j, k as $q \triangleq \begin{bmatrix} q_0 & q_v^T \end{bmatrix}^T \in \mathcal{S}^4$, where $\mathcal{S}^r \triangleq \left\{x \in \mathbb{R}^p | x^T x = 1\right\}$, and q_0 and q_v represent the scalar and vector components of q. Based on this definition, a vector expressed in the object coordinate system, $\xi_q \in \mathbb{R}^3$, can be related to the same vector expressed in the camera coordinate system, $\xi_c \in \mathbb{R}^3$, as $\xi_c = q \cdot \xi_q \cdot \bar{q}$, where $\bar{()} : \mathcal{S}^4 \to \mathcal{S}^4$ represents the unit quaternion inverse operator defined as $\bar{q} \triangleq \begin{bmatrix} q_0 & -q_v^T \end{bmatrix}^T$ with identity $\bar{q} \cdot q = q \cdot \bar{q} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$, and $(\cdot) : \mathbb{R}^4 \times \mathbb{R}^4 \to \mathbb{R}^4$ represents the Hamilton product[1], with property $q_a \cdot q_b \in \mathcal{S}^4$ for q_a, $q_b \in \mathcal{S}^4$. The Hamilton product can be expressed in block matrix notation as

$$q_a \cdot q_b = \begin{bmatrix} q_{a0} & -q_{av}^T \\ q_{av} & q_{a0} I_3 + q_{av}^{\times} \end{bmatrix} q_b$$

where $I_g \in \mathbb{R}^{g\times g}$ is the identity matrix. The kinematics for the relative orientation of the object with respect to the camera are (see [56, Chapter 3.4] or [57, Chapter 3.6])

$$\dot{q} = \frac{1}{2} B(q) (\omega_q - \bar{q} \cdot \omega_c \cdot q) \tag{2}$$

where $B : \mathcal{S}^4 \to \mathbb{R}^{4\times 3}$ is defined as

$$B(\xi) \triangleq \begin{bmatrix} -\xi_v^T \\ \xi_0 I_3 + \xi_v^{\times} \end{bmatrix}$$

and has the pseudoinverse property $B(\xi)^T B(\xi) = I_3$ (see [56, Chapter 3.4]).

III. Estimation Objective

The primary goal in this work is to develop a pose estimator/predictor that is robust to intermittent measurements. The design strategy is to filter the pose measurements when they are available, and predict future poses when measurements are unavailable (e.g., the object is not visible to the camera). However, a predictor based on (1) and (2) would require linear and angular velocities of the object to be known. The novelty in this work is to learn a model of the object velocities when measurements are available, and use the model in the predictor when measurements are not available. To this end, a stacked pose state, $\eta(t) \in \mathbb{R}^7$, is defined as $\eta(t) \triangleq \begin{bmatrix} x^T(t) & q^T(t) \end{bmatrix}^T$ and the following assumptions are utilized.

Assumption 1. Measurements of the relative pose of the target are available from camera images when the target is in view.

Remark 1. The projection of a 3D scene onto a 2D sensor during the imaging process results in scale ambiguity [58, Chapter 5.4.4]. In typical SfM observers, target velocity is used to inject scale into the system and recover the full Euclidean coordinates of the target. However, in the scenario considered in this paper, the target velocities are unknown. To resolve the ambiguity, a known length scale on the target can used, and by exploiting Perspective-n-Point (e.g., [59]–[65]) or homography (e.g., [66] and [67]) solvers, the pose of the target can be recovered.

Assumption 2. The object velocities are locally Lipschitz functions of the object pose and not explicitly time dependent, i.e., $v_q(t) = \phi_1(\rho(\eta(t), t))$ and $\omega_q(t) = \phi_2(\rho(\eta(t), t))$,

[1]For brevity, a slight abuse of notation will be utilized throughout the paper. For v_1, $v_2 \in \mathbb{R}^3$ and $q \in \mathcal{S}^4$, the equation $v_2 = q \cdot v_1 \cdot \bar{q}$ can be written precisely as $q_{v2} = q \cdot q_{v1} \cdot \bar{q}$, where $q_{v1} \triangleq \begin{bmatrix} 0 & v_1^T \end{bmatrix}^T$ and $q_{v2} \triangleq \begin{bmatrix} 0 & v_2^T \end{bmatrix}^T$. In other words, an $\mathbb{R}^4$ quaternion, q_{v1}, is derived from an $\mathbb{R}^3$ vector, v_1, by setting the scalar part of q_{v1} to zero and setting the vector part of q_{v1} as equal to v_1. Similarly, the resulting vector, v_2, is derived from the vector component of q_{v2}.

where ϕ_1, $\phi_2 : \mathbb{R}^7 \to \mathbb{R}^3$ are bounded and $\rho : \mathbb{R}^7 \times [0, \infty) \to \mathbb{R}^7$ is a known, bounded, and locally Lipschitz function.

Remark 2. This assumption ensures there exists some function that can be learned, i.e., the object velocities do not meander arbitrarily. Moreover, via the Stone–Weierstrass theorem [68], it ensures that universal function approximators (e.g., neural networks) can be used to estimate the object velocities to an arbitrary level of accuracy. The Stone–Weierstrass theorem only guarantees the estimate is accurate over a compact set, hence dependence on the state is allowed since it is bounded via Assumption 3 below, but exclusion of an explicit dependence on time is required since the interval $t \in [0, \infty)$ is considered in the analysis. The velocities can change with time, since the state of the object can change with time, however, the *mapping* between the object state and the object velocity is assumed to be static. This assumption holds in cases of e.g., projectile or orbital motion, pursuit-evasion games, as well as simplistic models of vehicles moving along a road network, e.g., the proof of concept experiments provided in Section VII.

Assumptions analogous to Assumption 2 are implicit in machine learning and function approximation contexts. Intuitively, if an explicit and unknown time dependence is allowed in the function to be estimated, there is no guarantee that the data used to approximate the function, and hence the function estimate, will be valid in the future. For example, in [47], the authors describe a scenario of tracking a target with a finite set of behaviors, and use a nonparametric approach to learn an anomalous behavior. This type of target motion could be learned using our approach if the velocity maps, ϕ_1 and ϕ_2, were piecewise-in-time static. For such a case, the analysis in the Section VI can be expanded to include switching due to changing target behavior. However, if the target exhibited new behavior (i.e., a new state in the Markov model) at every timestep, there would be no hope in learning the overall target behavior, since the past data would provide no insight into future behavior.

In some scenarios, information beyond the object pose (e.g., traffic levels, time of day, weather, etc.) may be relevant in predicting the target behavior. These auxiliary states can be considered in the function approximation to capture a wider class of possible target behavior without violating technical requirements underpinning learning. The auxiliary states can be included either directly if they are measurable, or by using an observer, hidden Markov model, etc. to generate state estimates if the the auxiliary states are not measurable.

Remark 3. In some applications, the velocity field of the target is expected to be dependent on the target's pose with respect to the world, rather than it's relative pose with respect to the camera. The function ρ is used to transform the relative pose to its world pose by using the camera pose with respect to the world. In other applications, the velocity field is expected to rely solely on the relative pose (e.g., a pursuit-evasion scenario in an obstacle free environment, where the evader's motion would only be dependent on it's pose with respect to the pursuer/camera) or the camera pose is unknown, in which case ρ can be taken as the identity function on $\eta(t)$. As shown in the following, these coordinate transformations are embedded in the bases of the function approximation.

Assumption 3. The state $\eta(t)$ is bounded, i.e. $\eta(t) \in \mathcal{X}$, where $\mathcal{X} \subset \mathbb{R}^7$ is a convex, compact set.

Remark 4. In estimation, for the state estimates to converge to the states while remaining bounded, the states themselves must remain bounded. This is analogous to the requirement of bounded desired trajectories in control problems.

In this development, the unknown motion model functions, ϕ_1 and ϕ_2, are approximated with a neural network, i.e.,

$$\begin{aligned} \begin{bmatrix} v_q(t) \\ \frac{1}{2}B(q(t))\,\omega_q(t) \end{bmatrix} &= \begin{bmatrix} \phi_1(\rho(\eta(t),t)) \\ \frac{1}{2}B(q(t))\,\phi_2(\rho(\eta(t),t)) \end{bmatrix} \\ &= W^T\sigma(\rho(\eta,t)) + \varepsilon(\rho(\eta,t)) \end{aligned} \tag{3}$$

where $\sigma : \mathbb{R}^7 \to \mathbb{R}^p$ is a known, bounded, locally Lipschitz, vector of basis functions, $W \in \mathbb{R}^{p\times 7}$ is a matrix of the unknown ideal weights, and $\varepsilon : \mathbb{R}^7 \to \mathbb{R}^7$ is the function approximation residual, which is locally Lipschitz based on the locally Lipschitz properties of $v_q(t)$, $\omega_q(t)$, $B(q(t))$, $\rho(\eta(t),t)$ and $\sigma(\cdot)$, and is *a priori* bounded with a bound that can be made arbitrarily small based on the Stone-Weierstrass theorem, i.e., $\bar{\varepsilon} \triangleq \sup_{\eta\in\mathcal{X},\, t\in[0,\infty)} \|\varepsilon(\rho(\eta,t))\|$, where $\|\cdot\|$ denotes the Euclidean norm. Note that if W is known, $\phi_2(\rho(\eta(t),t))$ can be approximated by premultiplying by $2B^T(q(t))$ and utilizing the pseudoinverse property of $B(q(t))$.

To quantify the estimation objective, let

$$\tilde{\eta}(t) \triangleq \eta(t) - \hat{\eta}(t) \tag{4}$$

denote the estimation error, where $\hat{\eta}(t) \in \mathbb{R}^7$ contains the position and orientation estimates. Also, let

$$\tilde{W}(t) \triangleq W - \hat{W}(t), \tag{5}$$

denote the parameter estimation error, where $\hat{W}(t) \in \mathbb{R}^{p\times 7}$ is the estimate of the ideal function approximation weights. Based on these definitions, the kinematics in (1) and (2) can be rewritten as

$$\dot{\eta}(t) = W^T\sigma(\rho(\eta(t),t)) + \varepsilon(\rho(\eta(t),t)) + f(\eta(t),t), \tag{6}$$

where $f : \mathbb{R}^7 \times [0,\infty) \to \mathbb{R}^7$ is a known function defined as

$$f(\eta(t),t) \triangleq -\begin{bmatrix} v_c(t) + \omega_c(t)^{\times} x(t) \\ \frac{1}{2}B(q(t))\,(\bar{q}(t)\cdot\omega_c(t)\cdot q(t)) \end{bmatrix}.$$

IV. Estimator Design

The following sections detail the estimator. The estimator is summarized in Algorithm 1, where δt refers to the loop timestep.

A. Update

Based on the subsequent stability analysis, during the periods in which measurements are available, the position and orientation estimate update laws are designed as

$$\begin{aligned} \dot{\hat{\eta}}(t) =& \hat{W}(t)^T\sigma(\rho(\eta(t),t)) + f(\eta(t),t) + k_1\tilde{\eta}(t) \\ &+ k_2\text{sgn}(\tilde{\eta}(t)), \end{aligned} \tag{7}$$

where $\text{sgn}(\cdot)$ is the signum function. To facilitate the design of an ICL update law, let $\mathcal{Y}(t) \triangleq \int_{t-\Delta t}^{t} \sigma^T(\rho(\eta(\tau),\tau))\,d\tau$ and

Algorithm 1 Algorithm for estimator

Input: $\hat{\eta}(0)$, $\hat{W}(0)$
Output: $\hat{\eta}(t)$, $\hat{W}(t)$

Initialization:
Initialize $\{t_1, ..., t_N\}$, $\{\mathcal{Y}_1, ..., \mathcal{Y}_N\}$, $\{\mathcal{F}_1, ..., \mathcal{F}_N\}$, $\{\Delta\eta_1, ..., \Delta\eta_N\}$ to 0
Estimator loop:
while target tracking **do**
  if target is in view **then**
    $\hat{\eta}(t) \leftarrow \int_{t-\delta t}^{t} \left[\hat{W}(\tau)^T \sigma(\rho(\eta(\tau),\tau)) + f(\eta(\tau),\tau) + k_1\tilde{\eta}(\tau) + k_2 \text{sgn}(\tilde{\eta}(\tau))\right] d\tau$
    $\hat{W}(t) \leftarrow \int_{t-\delta t}^{t} \left[\text{proj}\left(\Gamma\sigma(\rho(\eta(\tau),\tau))\tilde{\eta}(\tau)^T + k_{CL}\Gamma \sum_{i=1}^{N} \mathcal{Y}_i^T\left(\Delta\eta_i - \mathcal{F}_i - \mathcal{Y}_i\hat{W}(\tau)\right)\right)\right] d\tau$
    $\mathcal{Y}(t) \leftarrow \int_{t-\Delta t}^{t} \sigma^T(\rho(\eta(\tau),\tau))\, d\tau$
    $\mathcal{F}(t) \leftarrow \int_{t-\Delta t}^{t} f^T(\eta(\tau),\tau)\, d\tau$
    Data Selection:
    for $i = l$ to N **do**
      $\lambda_i \leftarrow \lambda_{\min}\left\{\mathcal{Y}(t)^T\mathcal{Y}(t) + \sum_{j=1,\, j\neq i}^{N} \mathcal{Y}_j^T\mathcal{Y}_j\right\}$
    end for
    $k \leftarrow \arg\max_i \{\lambda_i\}$
    if $\lambda_k > \lambda_{\min}\left\{\sum_{i=1}^{N} \mathcal{Y}_i^T\mathcal{Y}_i\right\}$ **then**
      $t_k \leftarrow t$
      $\mathcal{Y}_k \leftarrow \mathcal{Y}(t)$
      $\mathcal{F}_k \leftarrow \mathcal{F}(t)$
      $\Delta\eta_k \leftarrow \eta^T(t) - \eta^T(t-\Delta t)$
    end if
  else
    $\hat{\eta}(t) \leftarrow \int_{t-\delta t}^{t} \left[\text{proj}\left(\hat{W}(\tau)^T\sigma(\rho(\hat{\eta}(\tau),\tau)) + f(\hat{\eta}(\tau),\tau)\right)\right] d\tau$
    $\hat{W}(t) \leftarrow \int_{t-\delta t}^{t} \left[\text{proj}\left(k_{CL}\Gamma \sum_{i=1}^{N} \mathcal{Y}_i^T\left(\Delta\eta_i - \mathcal{F}_i - \mathcal{Y}_i\hat{W}(\tau)\right)\right)\right] d\tau$
  end if
end while
return $\hat{\eta}(t)$, $\hat{W}(t)$

$\mathcal{F}(t) \triangleq \int_{t-\Delta t}^{t} f^T(\eta(\tau),\tau)\, d\tau$, where $\Delta t \in \mathbb{R}$ is a positive constant denoting the size of the window of integration. The ICL update law for the motion model approximation parameters is designed as

$$\dot{\hat{W}} = \text{proj}\left(\Gamma\sigma(\rho(\eta(t),t))\tilde{\eta}(t)^T + k_{CL}\Gamma\sum_{i=1}^{N}\mathcal{Y}_i^T\left(\Delta\eta_i - \mathcal{F}_i - \mathcal{Y}_i\hat{W}(t)\right)\right), \quad (8)$$

where $\text{proj}(\cdot)$ is a smooth projection operator (see [69, Appendix E], [70, Remark 3.7]) with bounds based on the state bounds and velocity bounds of Assumptions 2 and 3, $N \in \mathbb{N}_0$, $k_{CL} \in \mathbb{R}$ and $\Gamma \in \mathbb{R}^{p\times p}$ are constant, positive definite and symmetric control gains, $\Delta\eta_i \triangleq \eta^T(t_i) - \eta^T(t_i - \Delta t)$, $\mathcal{F}_i \triangleq \mathcal{F}(t_i)$, $\mathcal{Y}_i \triangleq \mathcal{Y}(t_i)$, and t_i represents past time points, i.e., $t_i \in [\Delta t, t]$, at which measurements are available. The principal goal behind this design is to incorporate recorded input and trajectory data to identify the ideal weights. The time points t_i, and the corresponding $\Delta\eta_i$, $\mathcal{F}_i$, and $\mathcal{Y}_i$ that are recorded and used in (8) are referred to as the history stack. As shown in the subsequent stability analysis, the parameter estimate learning rate is related to the minimum eigenvalue of $\sum_{i=1}^{N}\mathcal{Y}_i^T\mathcal{Y}_i$, motivating the use of the singular value maximization algorithm in [54, Chapter 6] for adding or replacing data in the history stack.

To gain additional insight into the adaptive update law design in (8), the integral of the transpose of (6) is

$$\int_{t-\Delta t}^{t}\dot{\eta}^T(\tau)\, d\tau = \int_{t-\Delta t}^{t}\sigma^T(\rho(\eta(\tau),\tau))\, W\, d\tau + \int_{t-\Delta t}^{t}\varepsilon^T(\rho(\eta(\tau),\tau))\, d\tau + \int_{t-\Delta t}^{t} f^T(\eta(\tau),\tau)\, d\tau.$$

Using the Fundamental Theorem of Calculus and simplifying yields

$$\eta^T(t) - \eta^T(t-\Delta t) = \mathcal{Y}(t)\, W + \mathcal{E}(t) + \mathcal{F}(t), \ \forall t \in [\Delta t, \infty) \quad (9)$$

where, $\forall t \in [\Delta t, \infty)$, $\mathcal{E}(t) \triangleq \int_{t-\Delta t}^{t} \varepsilon^T(\rho(\eta(\tau),\tau))\, d\tau$. Using the relation in (9), the update law in (8) can be simplified as

$$\dot{\hat{W}} = \text{proj}\left(\Gamma \sigma(\rho(\eta(t),t))\,\tilde{\eta}(t)^T + k_{CL}\Gamma \sum_{i=1}^{N} \mathcal{Y}_i^T \mathcal{Y}_i \tilde{W}(t) + k_{CL}\Gamma \sum_{i=1}^{N} \mathcal{Y}_i^T \mathcal{E}_i\right), \tag{10}$$

for all $t > \Delta t$, where $\mathcal{E}_i \triangleq \mathcal{E}(t_i)$. Taking the time derivative of (4), substituting (6) and (7), and simplifying, yields the following closed-loop error dynamics when measurements are available

$$\dot{\tilde{\eta}}(t) = \tilde{W}(t)^T \sigma(\rho(\eta(t),t)) - k_1\tilde{\eta}(t) + \varepsilon(\rho(\eta(t),t)) - k_2 \text{sgn}(\tilde{\eta}(t)). \tag{11}$$

B. Predictor

During periods when measurements are not available, the state estimates are simulated forward in time using

$$\dot{\hat{\eta}}(t) = \text{proj}\left(\hat{W}(t)^T \sigma(\rho(\hat{\eta}(t),t)) + f(\hat{\eta}(t),t)\right). \tag{12}$$

Similarly, the recorded data continues to provide updates to the ideal weight estimates via

$$\dot{\hat{W}}(t) = \text{proj}\left(k_{CL}\Gamma \sum_{i=1}^{N} \mathcal{Y}_i^T\left(\Delta\eta_i - \mathcal{F}_i - \mathcal{Y}_i\hat{W}(t)\right)\right), \tag{13}$$

which can be simplified as

$$\dot{\hat{W}}(t) = \text{proj}\left(k_{CL}\Gamma \sum_{i=1}^{N} \mathcal{Y}_i^T \mathcal{Y}_i \tilde{W}(t) + k_{CL}\Gamma \sum_{i=1}^{N} \mathcal{Y}_i^T \mathcal{E}_i\right). \tag{14}$$

Taking the time derivative of (4), substituting (6) and (12), and simplifying yields the following closed-loop dynamics when measurements are not available

$$\dot{\tilde{\eta}}(t) = \tilde{W}(t)^T \sigma(\rho(\eta(t),t)) + f(\eta(t),t) - f(\hat{\eta}(t),t) + \hat{W}(t)^T(\sigma(\rho(\eta(t),t)) - \sigma(\rho(\hat{\eta}(t),t))) + \varepsilon(\rho(\eta(t),t)). \tag{15}$$

V. Implementation and Data Selection

The integration time window, Δt, can be selected small relative to the time scale of the dynamics (see Section VII for examples) to reduce the adverse effects of noise and function approximation error. After time Δt (i.e., $t > \Delta t$), the signals $\mathcal{Y}(t)$, $\mathcal{F}(t)$ and $\eta^T(t) - \eta^T(t-\Delta t)$ are available. For initialization of the history stack, the values of $\mathcal{Y}(t)$, $\mathcal{F}(t)$ and $\eta^T(t) - \eta^T(t-\Delta t)$ can be saved at every time step until N values have been recorded, and hence $\sum_{i=1}^{N} \mathcal{Y}_i^T\left(\Delta\eta_i - \mathcal{F}_i - \mathcal{Y}_i\hat{W}(t)\right)$ can be calculated. However, typically the data collected during initialization is not sufficiently rich (i.e., do not satisfy Assumption 4). Therefore, a procedure similar to that described in [54, Chapter 6] can be used for replacing data in the history stack. Specifically, if $\lambda_{\min}\left\{\mathcal{Y}(t)^T\mathcal{Y}(t) + \sum_{i=1,\, i\neq j}^{N} \mathcal{Y}_i^T\mathcal{Y}_i\right\} > \lambda_{\min}\left\{\sum_{i=1}^{N} \mathcal{Y}_i^T\mathcal{Y}_i\right\}$ for some $j \in \{1,2,...,N\}$, where $\lambda_{\min}\{\cdot\}$ refers to the minimum eigenvalue of $\{\cdot\}$, then replace t_j, $\mathcal{Y}_j$, $\mathcal{F}_j$, and $\Delta\eta_j$ with t, $\mathcal{Y}(t)$, $\mathcal{F}(t)$ and $\eta^T(t) - \eta^T(t-\Delta t)$, respectively. In this way, $\lambda_{\min}\left\{\sum_{i=1}^{N} \mathcal{Y}_i^T\mathcal{Y}_i\right\}$ is always increasing. If the system trajectories are sufficiently exciting (i.e., satisfy Assumption 4), $\lambda_{\min}\left\{\sum_{i=1}^{N} \mathcal{Y}_i^T\mathcal{Y}_i\right\}$ will be strictly greater than zero in finite time, at which point new data is not needed, and hence the system trajectories no longer need to be exciting.

VI. Analysis

The system considered in this work operates in two modes. The evolution of a Lyapunov-like function is developed in Lemma 1 for the mode when measurements are available and the update is used. Similarly, the evolution of a Lyapunov-like function is developed in Lemma 2 for the mode when measurements are unavailable and the predictor is active.

In addition to the switching that occurs as measurements become intermittently unavailable, in the following stability analysis, time is partitioned into two phases. During the initial phase, insufficient data has been collected to satisfy a richness condition on the history stack. In Theorem 1 it is shown that the designed estimator and adaptive update law are still sufficient for the system to remain bounded for all time despite the lack of data. After a finite period of time, the system transitions to the second phase, where the history stack is sufficiently rich and the estimator and adaptive update law are shown, in Theorem 2, to asymptotically converge to an arbitrarily small bound. To guarantee that the transition to the second phase happens in finite time, and therefore the overall system trajectories are ultimately bounded, we require the history stack be sufficiently rich after a finite period of time, as specified in the following assumption.

Assumption 4.

$$\exists \underline{\lambda},\ T > 0 : \forall t \geq T,\ \lambda_{\min}\left\{\sum_{i=1}^{N} \mathcal{Y}_i^T\mathcal{Y}_i\right\} \geq \underline{\lambda}, \tag{16}$$

where $\lambda_{\min}\{\cdot\}$ refers to the minimum eigenvalue of $\{\cdot\}$.

The condition in (16) requires that the system be sufficiently excited, though is weaker than the typical PE condition since excitation is only needed for a finite period of time. Specifically, PE requires

$$\int_{t}^{t+\Delta t} \sigma(\rho(\eta(\tau),\tau))\,\sigma^T(\rho(\eta(\tau),\tau))\, dt \geq \alpha I > 0,\ \forall t > 0 \tag{17}$$

whereas Assumption 4 only requires the system trajectories to be exciting up to time T (at which point $\sum_{i=1}^{N} \mathcal{Y}_i^T\mathcal{Y}_i$ is full rank), after which the exciting data recorded during $t \in [0, T]$ is exploited for all $t > T$. Another benefit of the development in this paper is that the excitation condition is measurable (i.e., $\lambda_{\min}\left\{\sum_{i=1}^{N} \mathcal{Y}_i^T\mathcal{Y}_i\right\}$ can be calculated), whereas in PE,

Δt is unknown, and hence an uncountable number of integrals would need to be calculated at each of the uncountable number of time points, t, in order to verify PE.

To facilitate the following analysis, let t_n^{on} and t_n^{off} denote the n^{th} instance at which measurements become available and unavailable, respectively. Then during $t \in [t_n^{\text{on}}, t_n^{\text{off}})$ measurements are available and the estimator is active, whereas during $t \in [t_n^{\text{off}}, t_{n+1}^{\text{on}})$ measurements are unavailable and the predictor is active. The duration of contiguous time each of these modes are active is denoted $\Delta t_n^{\text{on}} \triangleq t_n^{\text{off}} - t_n^{\text{on}}$ and $\Delta t_n^{\text{off}} \triangleq t_{n+1}^{\text{on}} - t_n^{\text{off}}$, respectively, and the total amount of time each of these modes is active between switching instances a and b are denoted $T^{\text{on}}(a,b) \triangleq \sum_{i=a}^{b} \Delta t_i^{\text{on}}$ and $T^{\text{off}}(a,b) \triangleq \sum_{i=a}^{b} \Delta t_i^{\text{off}}$, respectively. Also, $\xi(t) \triangleq \begin{bmatrix} \tilde{\eta}(t)^T & \text{vec}\left(\tilde{W}(t)\right)^T \end{bmatrix}^T \in \mathbb{R}^{7+7p}$ denotes a stacked state and parameter error vector, where $\text{vec}(\cdot)$ denotes a stack of the columns of $(\cdot)$.

To facilitate the Lyapunov-based analysis in Lemmas 1 and 2, as well as Theorems 1 and 2, consider the Lyapunov function candidate $V : \mathbb{R}^{7+7p} \to \mathbb{R}$ defined as

$$V(\xi(t)) \triangleq \frac{1}{2}\tilde{\eta}(t)^T \tilde{\eta}(t) + \frac{1}{2}\text{tr}\left(\tilde{W}(t)^T \Gamma^{-1} \tilde{W}(t)\right). \quad (18)$$

The function in (18) can be bounded as $\beta_1 \|\xi(t)\|^2 \leq V(\xi(t)) \leq \beta_2 \|\xi(t)\|^2$, where $\text{tr}(\cdot)$ denotes the matrix trace operator, $\beta_1 \triangleq \frac{1}{2}\min\left\{1, \lambda_{\min}\left(\Gamma^{-1}\right)\right\}$, and $\beta_2 \triangleq \frac{1}{2}\max\left\{1, \lambda_{\max}\left(\Gamma^{-1}\right)\right\}$. Also, due to the projection operator in (8) and (13), and since W is a constant, $\tilde{W}(t)$ is bounded, and $V(\xi(t)) \leq c_2 + c_3 \|\tilde{\eta}(t)\|^2$, where $c_2,\ c_3 \in \mathbb{R}_{>0}$ are positive constants.

Lemma 1. *The estimator in (7) and (8) remains bounded during $t \in [t_n^{\text{on}}, t_n^{\text{off}})$.*

Proof: Taking the time derivative of (18) during $t \in [t_n^{\text{on}}, t_n^{\text{off}})$, substituting (10) and (11), and simplifying yields

$$\dot{V}(\xi(t)) \leq -k_1 \|\tilde{\eta}(t)\|^2 + c_1,$$

where $c_1 \in \mathbb{R}_{>0}$ is a positive constant. Using the bounds on V, $\dot{V}$ can be bounded as

$$\dot{V}(\xi(t)) \leq -\frac{k_1}{c_3} V(\xi(t)) + \left(\frac{k_1 c_2 + c_1}{c_3}\right).$$

Using the Comparison Lemma [71, Lemma 3.4],

$$V(\xi(t)) \leq V(\xi(t_n^{\text{on}}))\exp\left[-\lambda(t - t_n^{\text{on}})\right] + \left(c_2 + \frac{c_1}{k_1}\right) \quad (19)$$

$\forall t \in [t_n^{\text{on}}, t_n^{\text{off}})$, where $\lambda \triangleq \frac{k_1}{c_3}$.

After sufficient data has been gathered (i.e., $t \in [t_n^{\text{on}}, t_n^{\text{off}}) \cap [T, \infty)$, where T was defined in Assumption 4)

$$V(\xi(t)) \leq V(\xi(t_n^{\text{on}}))\exp\left[-\lambda_T(t - t_n^{\text{on}})\right] + c_{UB} \quad (20)$$

where $\lambda_T \triangleq \frac{\min\{k_1, \lambda_{CL}\}}{\beta_2}$, $c_{UB} \triangleq \frac{c_1 \beta_2}{\min\{k_1, \lambda_{CL}\}}$, $\lambda_{CL} \triangleq k_{CL}\lambda_{\min}\left\{\sum_{i=1}^{N} \mathcal{Y}_i^T \mathcal{Y}_i\right\}$, and $\lambda_{CL} > 0$ based on Assumption 4. ∎

Remark 5. Note that c_1 is based on a bound on the data in the history stack, $\mathcal{Y}_i$, the concurrent learning gain, k_{CL}, and the bound on function approximation error, $\bar{\varepsilon}$, and therefore cannot be arbitrarily decreased through gain tuning. However, the ultimate error bound after sufficient data has been gathered, c_{UB}, can be made arbitrarily small by increasing the gains k_1 and k_{CL}, and by decreasing $\bar{\varepsilon}$, e.g., increasing the number of neurons in the NN.

Lemma 2. *The predictor in (12) and (13) remains bounded during $t \in [t_n^{\text{off}}, t_{n+1}^{\text{on}})$.*

Proof: Taking the time derivative of (18) during $t \in [t_n^{\text{off}}, t_{n+1}^{\text{on}})$, substituting (14) and (15), and simplifying yields

$$\dot{V}(\xi(t)) \leq c_4 \|\xi(t)\|^2 + c_5$$

where $c_4,\ c_5 \in \mathbb{R}_{>0}$ are positive constants. Using bounds on V, $\dot{V}$ can be bounded as

$$\dot{V}(\xi(t)) \leq \frac{c_4}{\beta_1} V(\xi(t)) + c_5.$$

Using the Comparison Lemma [71, Lemma 3.4],

$$V(\xi(t)) \leq V\left(\xi\left(t_n^{\text{off}}\right)\right)\exp\left[\frac{c_4}{\beta_1}\left(t - t_n^{\text{off}}\right)\right],\ \forall t \in [t_n^{\text{off}}, t_{n+1}^{\text{on}}) \quad (21)$$

which remains bounded for all bounded t. ∎

Theorem 1. *The estimator and predictor in (7), (8), (12), and (13) remain bounded provided there exists a $k < \infty$, and sequences $\{\Delta t_n^{\text{on}}\}_{n=0}^{\infty}$ and $\{\Delta t_n^{\text{off}}\}_{n=0}^{\infty}$ such that*

$$\frac{c_4}{\beta_1} T^{\text{off}}(nk, (n+1)k) < \lambda_T T^{\text{on}}(nk, (n+1)k),\ \forall n \in \mathbb{N}. \quad (22)$$

Proof: Consider a single cycle of losing and regaining measurements, i.e., $t \in [t_n^{\text{on}}, t_{n+1}^{\text{on}})$. Based on (19) and (21)

$$V\left(\xi\left(t_{n+1}^{\text{on}}\right)\right) \leq V(\xi(t_n^{\text{on}}))\exp\left[\frac{c_4}{\beta_1}\Delta t_n^{\text{off}} - \lambda \Delta t_n^{\text{on}}\right] + \left(c_2 + \frac{c_1}{k_1}\right)\exp\left[\frac{c_4}{\beta_1}\Delta t_n^{\text{off}}\right]. \quad (23)$$

Using (23), the evolution of V over k cycles is

$$V\left(\xi\left(t_{(n+1)k}^{\text{on}}\right)\right) \leq c_6 V(\xi(t_{nk}^{\text{on}})) + c_7$$

where $c_6,\ c_7 \in \mathbb{R}_{>0}$ are positive, bounded constants, and $c_6 < 1$ based on (22). Let $\{s_n\}_{n=0}^{\infty}$ be a sequence defined by the recurrence relation

$$s_{n+1} = M(s_n),$$

with initial condition $s_0 = V(\xi(t_0^{\text{on}}))$, where $M : \mathbb{R} \to \mathbb{R}$ is defined as $M(s) \triangleq c_6 s + c_7$. Since $c_6 < 1$, M is a contraction [72, Definition 9.22], and therefore all initial conditions, s_0, approach the fixed point $s = \frac{c_7}{1 - c_6}$ [72, Theorem 9.23]. Since the sequence $\{s_n\}$ upper bounds V in the sense that $V(\xi(t_{nk}^{\text{on}})) \leq s_n$, V is also ultimately bounded. However, V may grow within $\left[t_{nk}^{\text{on}}, t_{(n+1)k}^{\text{on}}\right]$ since the dwell time condition in (22) is specified over k cycles rather than a single cycle, and therefore the ultimate bound of ξ, which is based on the ultimate bound of V, is

$$\limsup_t \|\xi(t)\| \leq \beta_1 \frac{c_7}{1 - c_6}\exp\left(\frac{c_4}{\beta_1} T_{\max}^{\text{off}}\right),$$

where $T^{\text{off}}_{\max} \triangleq \sup_n T^{\text{off}}(nk,(n+1)k)$. ■

Theorem 2. *After sufficient data is collected, i.e., $t \in [T,\infty)$, the estimator and predictor in (7), (8), (12), and (13) converge to a bound that can be made arbitrarily small provided there exists a $k < \infty$, and sequences $\{\Delta t^{\text{on}}_n\}_{n=0}^{\infty}$ and $\{\Delta t^{\text{on}}_n\}_{n=0}^{\infty}$ such that (22) is satisfied.*

Proof: The proof follows similarly to the proof of Theorem 1. Consider a single cycle of losing and regaining measurements after sufficient data has been collected, i.e., $t \in [t^{\text{on}}_n, t^{\text{on}}_{n+1}) \cap [T,\infty)$. Based on (20) and (21)

$$V\left(\xi\left(t^{\text{on}}_{n+1}\right)\right) \leq V\left(\xi\left(t^{\text{on}}_n\right)\right)\exp\left[\frac{c_4}{\beta_1}\Delta t^{\text{off}}_n - \lambda_T \Delta t^{\text{on}}_n\right] + c_{UB}\exp\left[\frac{c_4}{\beta_1}\Delta t^{\text{off}}_n\right]. \tag{24}$$

Using (24), the evolution of V over k cycles is

$$V\left(\xi\left(t^{\text{on}}_{(n+1)k}\right)\right) \leq c_8 V\left(\xi\left(t^{\text{on}}_{nk}\right)\right) + c_9$$

where c_8, $c_9 \in \mathbb{R}_{>0}$ are positive, bounded constants, and $c_8 < 1$ based on (22). By using the same contraction arguments as in Theorem (1), the ultimate bound of ξ, is

$$\limsup_t \|\xi(t)\| \leq \beta_1 \frac{c_9}{1-c_8}\exp\left(\frac{c_4}{\beta_1}T^{\text{off}}_{\max}\right),$$

where $T^{\text{off}}_{\max} \triangleq \sup_n T^{\text{off}}(nk,(n+1)k)$. ■

Remark 6. The fundamental difference between Theorem 1 and Theorem 2, and hence the need for sufficiently rich data, is the control over the ultimate error bound. In Theorem 1, c_7 is based on c_2, which is based on the projection bound on the ideal function approximation weight errors, which is *a priori* determined, and therefore the ultimate error bound cannot be decreased. In Theorem 2, c_9 is based on c_{UB} which can be made arbitrarily small by, for example, increasing the number of neurons in the NN.

Remark 7. The dwell time condition in (22) is similar to an average dwell time condition, but only over k cycles. The condition requires that, over k cycles, the total amount of time that the stable subsystem is active (i.e., the target in view), scaled by the decay rate of the stable subsystem, is greater than the total amount of time that the unstable subsystem is active (i.e., the target not in view), scaled by the error growth rate. This is a relaxed condition compared to typical (i.e., single cycle) forward and reverse dwell time conditions as it allows flexibility in allocating time in the subsystems over k cycles. For example, if a large amount of time is spent observing the target in the first of k cycles, relatively little time is needed with the target in view in the remaining $k-1$ cycles to still satisfy (22) and ensure error convergence. With single cycle dwell time conditions, any surplus time spent observing the target beyond what is necessary to satisfy the dwell time condition has no benefit in the sense of relaxing the dwell time requirements of subsequent cycles.

VII. Experiments

Experiments were performed to verify the theoretical results and demonstrate the performance of the developed estimation and prediction scheme with online model learning. In the first experiment (Section VII-A), a stationary camera observed a target moving along a smooth vector field. In the second experiment (Section VII-B), a moving camera observed two targets moving along a road network. In both experiments, a Clearpath Robotics TurtleBot 2 with a Kobuki base was utilized as a mobile vehicle simulant (i.e., the target). A fiducial marker was mounted on the mobile robot, and a corresponding tracking software library (see [73] and [74]) was used to repeatably track the image feature pixel coordinates, as well as provide target pose measurements, when the target was in the camera FOV. A NaturalPoint, Inc. OptiTrack motion capture system was used to record the ground truth pose of the camera and target at a rate of 360 Hz. The pose provided by the motion capture system was also used to estimate the linear and angular velocities of the camera necessary for the estimator, where the current camera velocity estimates were taken to be the slope of the linear regression of the 20 most recent pose data points. The same procedure was used to calculate the linear and angular velocities of the target for ground truth and comparison with the learned model.

For both experiments, radial basis functions (RBF) were used in the NN, with parameters selected based on the description provided in the subsequent sections. Estimator gains were selected as $k_1 = 3$, $k_2 = 0.1$, $k_{CL} = 1$ and $\Gamma = I$, and the integration window was selected as $\Delta t = 0.1$ s. Further discussion on how the parameters and gains were selected is provided in Section VII-C.

A. Target Motion Along a Smooth Vector Field

In the first experiment, a stationary camera observed a target moving in a vector field of the form shown in Figure 2. An IDS UI-3060CP camera was used to capture 1936x1216 pixel resolution images at a rate of 60 frames per second. The function $\rho(\eta,t)$ introduced in Assumption 2 was used to determine the estimated 2D position of the target in the world coordinate system using the camera pose (see Remark 3). For this experiment, 81 kernels were used in the NN, with means arranged in a uniform 9x9 grid across the vector field (see Figure 2) and covariance selected as $\Sigma_k = 0.3I_2$. A total of $N = 600$ data points were saved in the CL history stack. During the first 60 seconds of the experiment, target visibility was maintained to quickly fill the CL history stack. Data was added at a rate of approximately 1 sample per second (the rate at which the data selection algorithm described in Section V could be executed), resulting in 10% of the history stack filled at the end of the initial learning phase. After the initial phase, periodic measurement loss was induced artificially by intermittently disregarding pose measurements and switching to the predictor. The dwell times for each period were selected randomly as $\Delta t^{\text{on}}_n \sim U(15,30)$ and $\Delta t^{\text{off}}_n \sim U(10,20)$. The results of this experiment are shown in Figures 3 through 7. As shown in Figures 3 and 4, the predictor initially performs poorly; however, prediction significantly improves as more

data is acquired. Boundedness of the unknown parameter estimates is validated in Figure 5. Figures 6 and 7 demonstrate that once sufficient data is acquired, the NN output tracks the motion of the target well, therefore reducing the need for large feedback and sliding mode gains, as well as accurately predicting target motion when measurements are unavailable. Accurate motion prediction is achieved despite the target deviating from the prescribed vector field due to the nonholonomic constraints on the mobile robot, as well as random disturbances such as wheel slip. The norm of the position RMSE vector was 0.25m and the orientation RMSE was 24.4° for this experiment, considering data after 200s (i.e., after initial data collection).

For comparison, a modified version of the estimator was implemented on the same data collected during the first experiment. The modification represents the case of using an ideal predictor, analogous to one of the scenarios considered in [48]. Specifically, the feedforward NN terms were replaced with the actual target velocities in this modification, leading to the position estimates shown in Figure 8. Despite the improved performance, the target velocities are typically not available, and therefore this design may not be implementable in many applications. However, through the learning scheme developed in this paper, the estimator performance quickly approaches that of the ideal scenario, without requiring target motion information. For comparison, the norm of the position RMSE vector was 0.03m and the orientation RMSE was 1.2° for this experiment using the ideal predictor, confirming the obvious notion that perfect velocity prediction results in better performance.

An extended Kalman filter (EKF) with a constant velocity model was also tested for comparison. The measurement data and visibility times were the same as in the first experiment, resulting in the position estimates shown in Figure 9. As expected, the EKF performs well when the actual target velocities are constant, in comparison to our estimator during the beginning of the experiment, since our estimator is still learning the motion model. However, as the target velocity changes, as would be common in many practical scenarios, our model-learning approach clearly outperforms the EKF, even in stretches where the target velocity is constant. One way to improve the initial performance of our estimator (e.g., while the model parameters are still being learned) would be to estimate the position and velocity of the target, as is done in the EKF. However, the augmented state space would require a larger NN, and therefore more data, and it is not clear that an acceleration motion model would be superior to a velocity motion model. With the EKF, the norm of the position RMSE vector was 1.67m and the orientation RMSE was 57.5° for this experiment, both of which are much larger than the corresponding errors using our model learning approach.

B. Multiple Targets on a Road Network

A second experiment was performed to demonstrate the utilization of results developed in this paper to an application. Specifically, the goal of this experiment was to use a single moving camera to estimate the pose of two targets

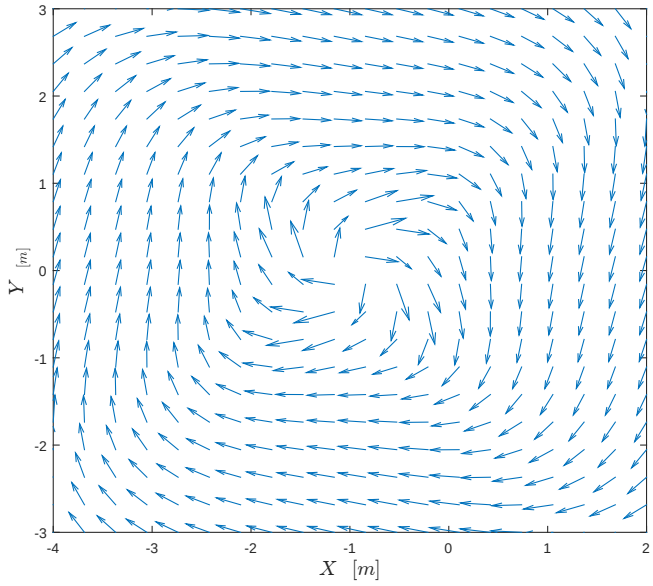

Figure 2. During the first experiment, the target was commanded to follow a vector field of this form.

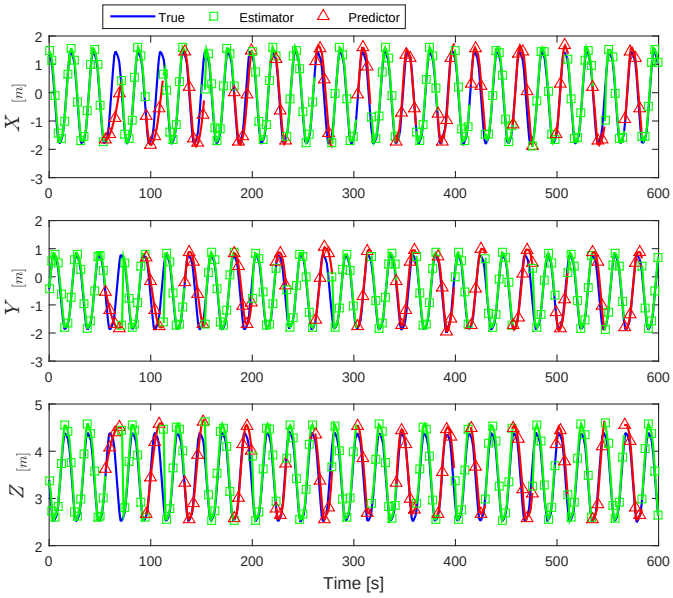

Figure 3. Relative position estimates for the first experiment with a stationary camera.

independently moving along an unknown road network (shown in Figures 10 and 11). At intersections in the road network, the targets randomly selected a direction to travel, hence violating Assumption 2. In this experiment, a camera on-board a Parrot Bebop 2 quadcopter platform was used to capture 640x368 pixel resolution images (see Figure 12), which were wirelessly streamed to an off-board computer at 30 frames per second. The function $\rho(\eta, t)$ was augmented to also output the estimated target heading in the world coordinate system, and the NN was composed of 172 kernels, with mean positions evenly spaced along the roads at 0.3 m intervals, mean headings parallel to the road, and covariance $\Sigma_k = 0.1I_3$. Two

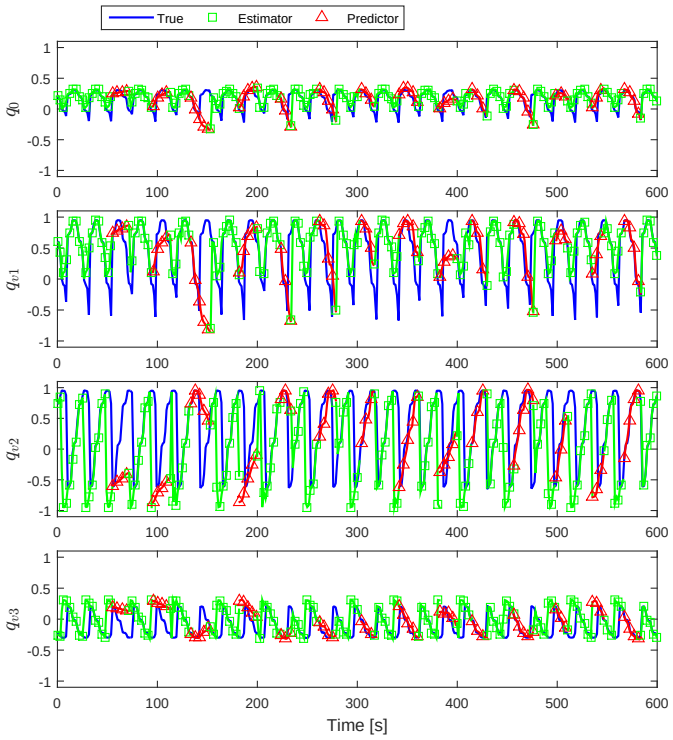

Figure 4. Relative orientation estimates for the first experiment with a stationary camera.

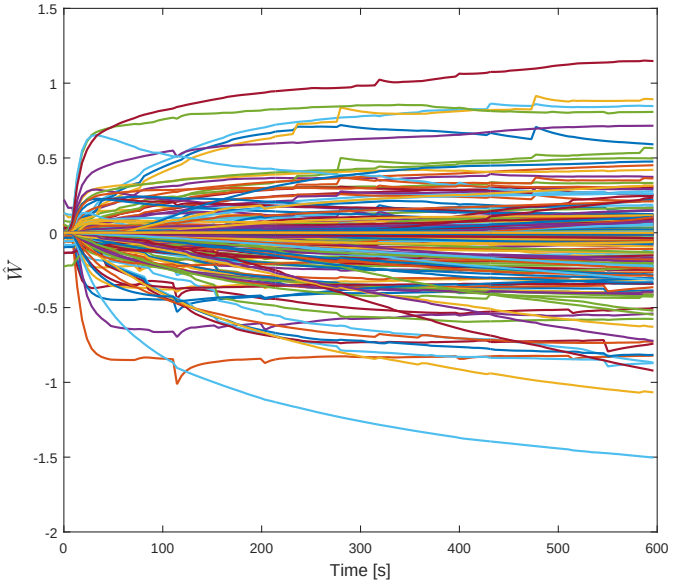

Figure 5. Evolution of the NN ideal weight estimates during the first experiment with a stationary camera.

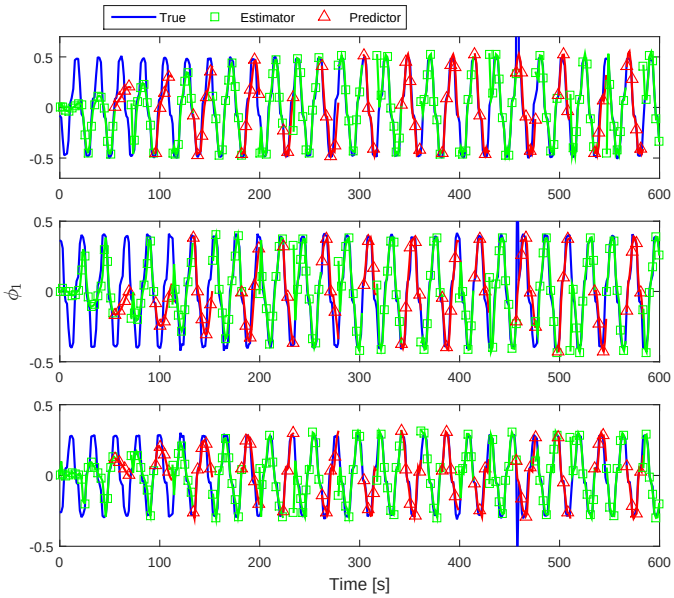

Figure 6. Output of the NN compared with ground truth linear velocities for the first experiment with a stationary camera. The jump in the ground truth signal at approximately 450s was caused by inaccurate numerical velocity approximation when the motion capture system temporary lost track of the target.

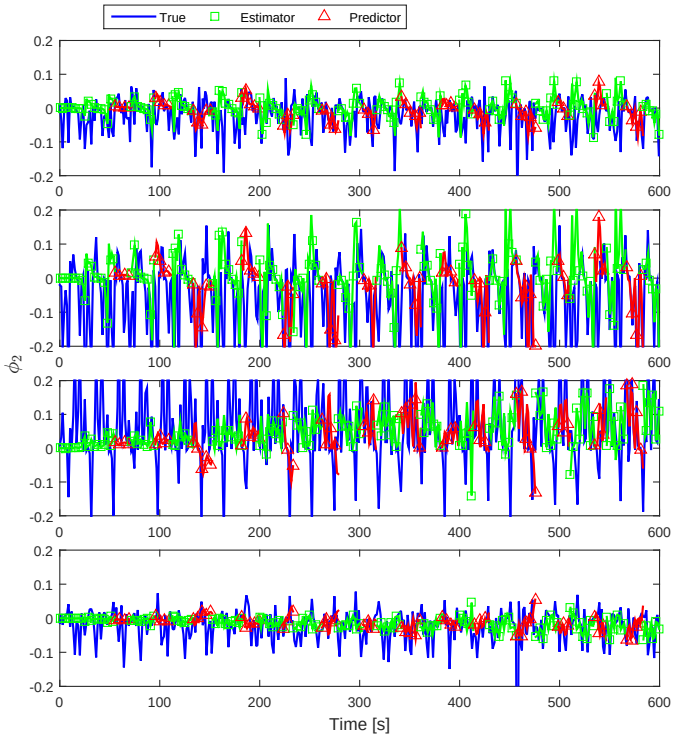

Figure 7. Output of the NN compared with ground truth orientation rates (i.e., $\frac{1}{2}B\left(q\left(t\right)\right)\omega_q\left(t\right)$) for the first experiment with a stationary camera.

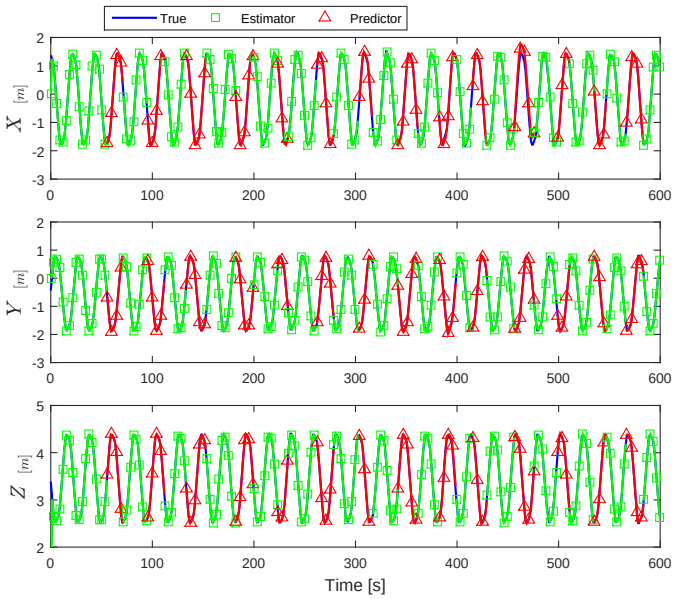

Figure 8. Relative position estimates for the experiment with a stationary camera using a modified observer with an ideal predictor. Although performance is satisfactory, this design requires unmeasurable target velocity information, and therefore is not implementable in many applications.

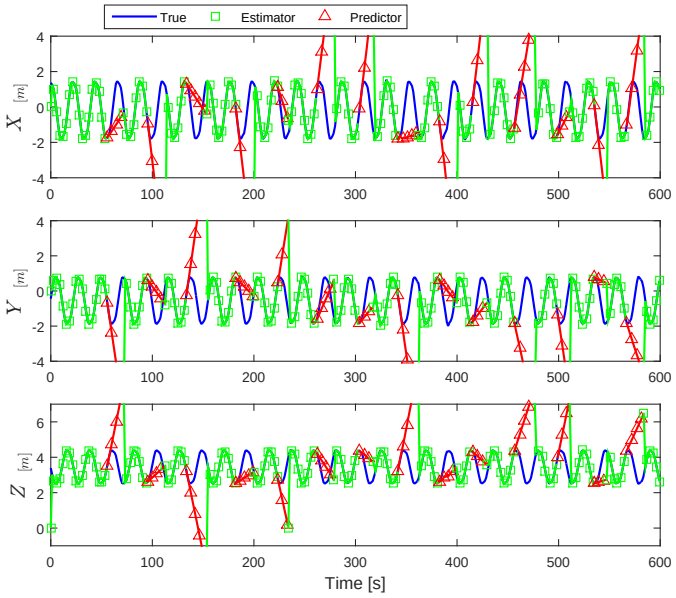

Figure 9. Relative position estimates for the experiment with a stationary camera using a Extended Kalman Filter with a constant velocity model. Without a good motion model, the EKF quickly diverges when the target is not in view.

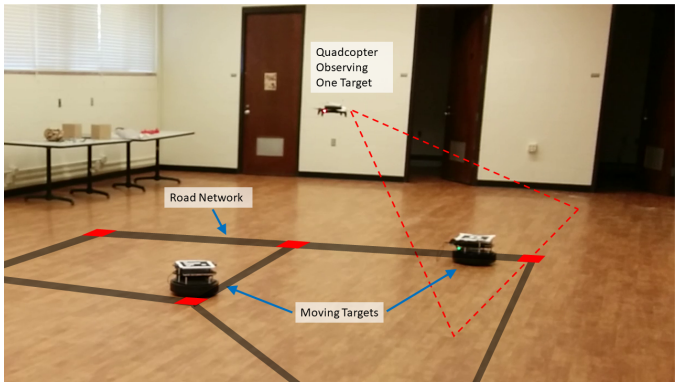

Figure 10. Overall setup of the second experiment, with quadcopter observer the targets as they move along a road network.

independent instances of the estimator developed in this paper were used to estimate the target poses, one for each target; however, for simplicity, since the targets share a common road network, the CL history stack was shared between the two estimators, with a total of $N = 2000$ data points saved in the stack. Independent history stacks could also have been used. During the initial phase, the quadcopter was commanded to follow a single target for approximately 300 seconds, therefore acquiring enough data to reasonably approximate a motion model of the targets along the road network. After the initial phase, the quadcopter was commanded to follow whichever target was closest to an intersection, since this is where the assumptions are violated, i.e., a deterministic function approximator would not be expected to accurately approximate a stochastic function. After the target selected a direction, and left the intersection, the predictor for this target is activated, and the quadcopter follows the other target. This strategy matches a reasonable strategy one might employ in a real world scenario: observe a target at intersections or other areas where the target can act randomly, but once the target has selected a direction, a sufficiently learned predictor is expected to perform well, and the observer can move on to other targets.

The results of this experiment are shown in Figures 13 through 16, and a video demonstrating the experiment is available at https://www.youtube.com/watch?v=QCIQtsQdhsM. Figures 13-16 show the true and estimated pose of the targets in world coordinates, thus demonstrating that after sufficient data is collected, the target pose can be accurately estimated even if the target remains outside the camera FOV for significant durations, despite significant delay due to the wireless transmission of the images, as well as the decreased measurement accuracy compared to the first experiment due to the low resolution camera.

C. Parameter Selection

As with many function approximation techniques, the parameters used for the NN are dependent on the specific application. For the experiments discussed in the preceding sections, commonly used RBFs were selected as the kernel since they exhibit local similarity (i.e., for the experiments, nearby points in the state space are expected to have similar

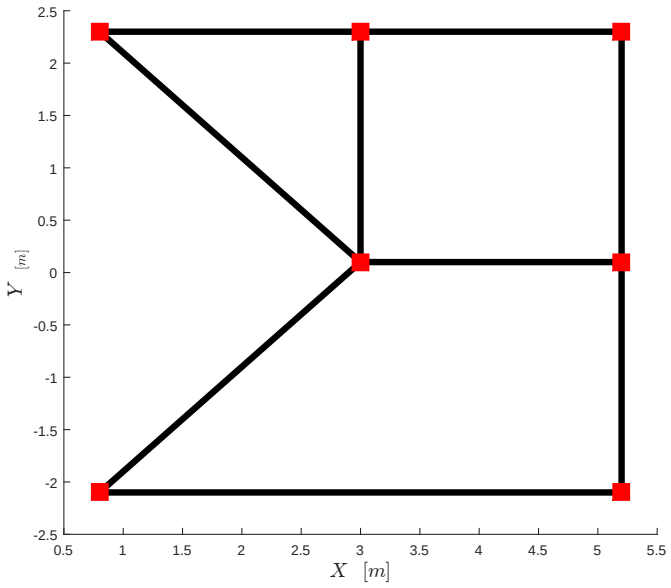

Figure 11. During the second experiment, both targets traveled along this network, randomly selecting turns at intersections.

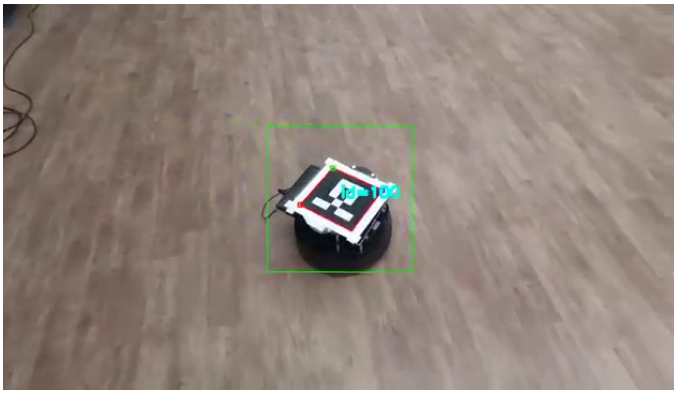

Figure 12. View from the flying quadcopter. Only one target is visible in the field of view at a time.

velocity values, and RBFs have increasingly similar activation for increasingly similar inputs). As demonstrated by the results, RBFs performed satisfactorily, and therefore, more exotic kernels were not considered, but could be explored for other applications as necessary.

The primary concern for selecting the number and distributions of the kernels is to ensure the relevant parts of the state space (i.e., the areas where the targets are expected to move through) have non-zero kernel activation. The secondary concern is to match the density and parameters of the kernels (e.g., mean and variance for RBFs) to the complexity of the underlying vector field to be approximated. In other words, regions of the vector field that are expected to have large spatial derivatives should be approximated with a dense distribution of kernels, each with relatively little extent (low variance for RBFs). If little is known about where the target may travel or how aggressively it may maneuver, a conservative approach can be taken, where a very large number of kernels can be distributed over a large section of the state space, and then

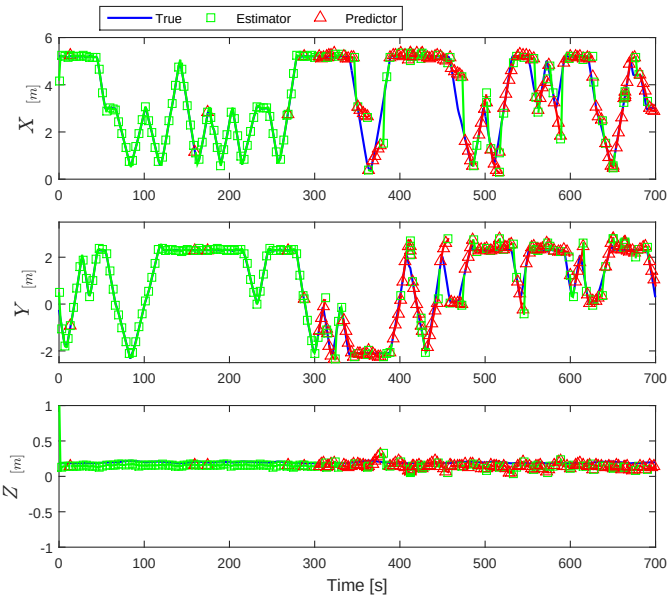

Figure 13. Position estimates of target 1 expressed in world coordinates for the second experiment with a quadcopter observing two moving targets.

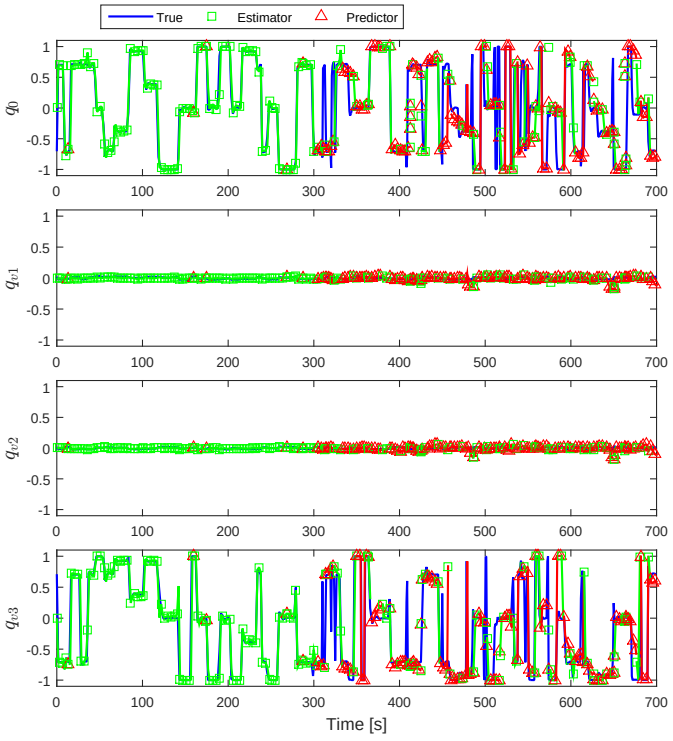

Figure 14. Orientation estimates of target 1 relative to the world coordinate system for the second experiment with a quadcopter observing two moving targets.

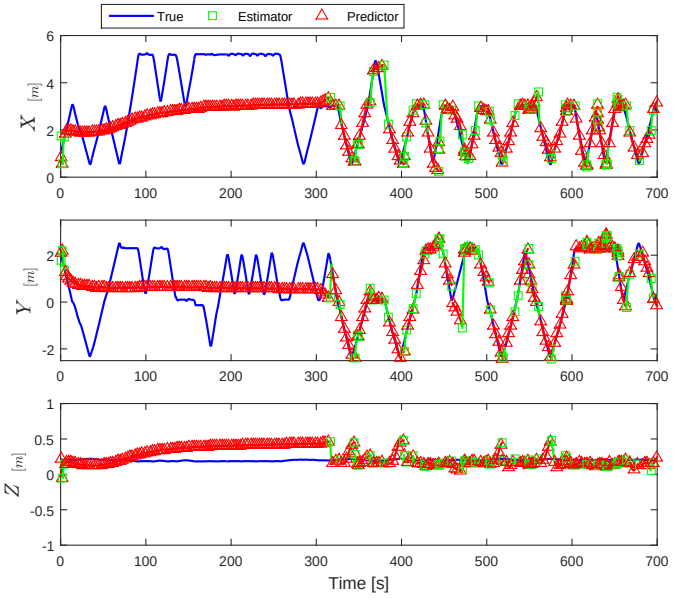

Figure 15. Position estimates of target 2 expressed in world coordinates for the second experiment with a quadcopter observing two moving targets.

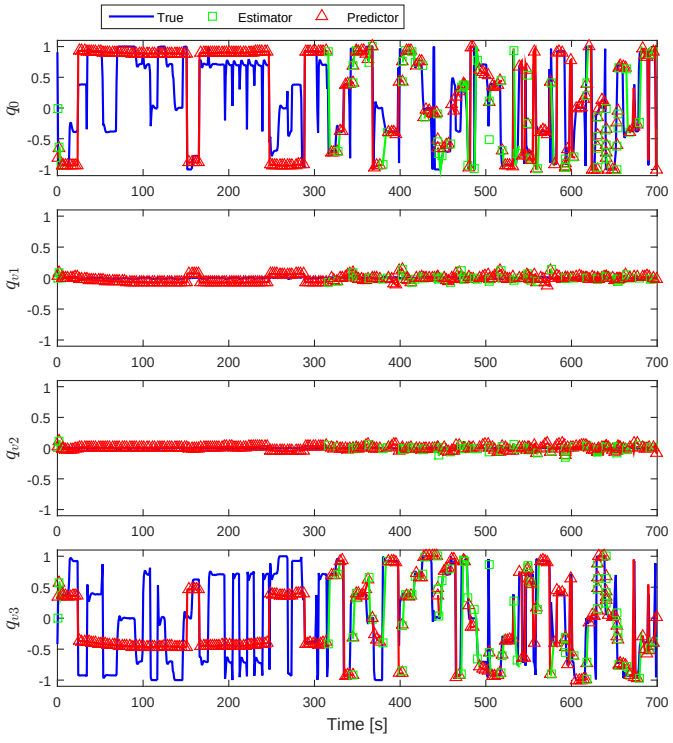

Figure 16. Orientation estimates of target 2 relative to the world coordinate system for the second experiment with a quadcopter observing two moving targets.

packing the kernels densely and with tight spatial extent. It is not surprising that with no knowledge of the operational space, that the resulting conservative approach would have increased memory and computational requirements.

In each experiment, the centers of the RBFs were distributed across the vector field and road network, respectively. For the second experiment, where the output of ρ also included the target heading, the kernel centers were doubled, one for each of the two directions parallel to the road. Kernel centers were separated by a distance approximately 0.3 m and had variance of $0.3I_2$ and $0.1I_3$ for the first and second experiment, respectively, where the smaller covariance was selected for the second experiment due to the tight turns in the road map. In both experiments, these initial parameter values performed satisfactorily, suggesting this approach is insensitive to NN tuning.

The two experiments use two different imaging sensors with varying capabilities. In the first experiment, a stationary high resolution camera is used to capture images at a high frame rate, and images are transferred over a wired connection with low latency and without compression. In the second experiment, a moving low resolution camera is used to capture images at a relatively lower frame rate, and the images are transferred over a wireless connection with high latency and lossy compression. The experiments demonstrate the viability of our approach in both cases using almost identical estimator parameters (with minor differences in the NN kernel parameters, as described previously), suggesting insensitivity to estimator gains despite e.g., frame drops, delay, lower measurement accuracy due to the lower image resolution, etc.

Minor tuning of the gains (k_1, k_2, k_{CL} and Γ) was required beyond the initial values of 1 or I.. Since signum functions are known to produce high frequency chatter, k_2 was reduced to 0.1, and k_1 was increased to 3.0 to compensate for the reduced robustness and increase the estimate convergence rate when measurements were available. The integration time window, Δt, was selected to be approximately equal to the timescale of changes in the target velocity so as not to "average out" complex target behavior. Our initial selection of $\Delta t = 0.1$ resulted in satisfactory estimator performance, and therefore no additional tuning was performed.

D. Discussion

Beyond the restriction on the target behavior to ensure learning is possible based on Assumption 2, the preceding experiments demonstrate that the remaining assumptions are either easily satisfied or do not significantly hinder estimator performance when not met. As discussed in Remark 1, Assumption 1 (i.e., availability of pose measurements) can be satisfied by currently available computer vision techniques and minimal domain knowledge. Assumption 3 (boundedness of the target pose) is easily satisfied in any practical scenario. Assumption 4 (sufficient richness in the data) is required to ensure NN parameter estimate convergence; however, predictor performance may be sufficient without it. For example, in the preceding experiments, the calculated minimum eigenvalue of the history stack remained within the floating point precision

floor of the computer system, suggesting the history stack is not full rank. Despite that, Figures 6 and 7 demonstrate satisfactory predictor performance.

VIII. Conclusion

An adaptive observer and predictor were developed to estimate the relative pose of a target from a camera in the presence of intermittent measurements. While measurements are available, data is recorded and used to update an estimate of the target motion model. When measurements are not available, the motion model is used in a predictor to update state estimates. The overall framework is shown to yield ultimately bounded estimation errors, where the bound can be made arbitrarily small through gain tuning, increasing data richness, and function approximation tuning. Experimental results demonstrate the performance of the developed estimator, even in cases of stochastic target motion where the assumptions are violated.

Although the experiments demonstrate a robustness to moderate violations of Assumption 2 (i.e., the targets in the experiment with the road network do not always follow a static vector field), estimation and prediction where a model does not exist (e.g., the target follows a non-periodic, time-varying trajectory) is still an open problem. However, it may be possible to use the methods developed in this paper to track objects undergoing a wider class of motions through relaxation of Assumption 2. First, states beyond just the object pose (e.g., traffic levels) can be used in the neural network to predict the object velocities. This can be used to account for varying object behavior without relying on explicit time dependence. Further, object velocity models can be expanded to allow for piecewise constant mappings between the object state and object velocity. This way, data collected to estimate the model is ensured to be valid for at least a finite period of time, while divergence of the measurements from prediction can be used as an indication that new data needs to be collected, or that other components of the system (e.g., feature tracking) are producing erroneous output. This approach could allow the tracked objects to switch their mode of operation.

References

[1] 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.

[2] O. Dahl, Y. Wang, A. F. Lynch, and A. Heyden, "Observer forms for perspective systems," *Automatica*, vol. 46, no. 11, pp. 1829–1834, Nov. 2010.

[3] 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. Am. Control Conf.*, Fairmont Queen Elizabeth, Montreal, Canada, Jun. 2012, pp. 4116–4123.

[4] 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.

[5] R. Spica, P. R. Giordano, and F. Chaumette, "Active structure from motion: Application to point, sphere, and cylinder," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1499–1513, Dec. 2014.

[6] A. Dani, N. Fischer, Z. Kan, and W. E. Dixon, "Globally exponentially stable observer for vision-based range estimation," *Mechatron.*, vol. 22, no. 4, pp. 381–389, Special Issue on Visual Servoing 2012.

[7] W. E. Dixon, Y. Fang, D. M. Dawson, and T. J. Flynn, "Range identification for perspective vision systems," *IEEE Trans. Autom. Control*, vol. 48, pp. 2232–2238, 2003.

[8] 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, 2005.

[9] X. Chen and H. Kano, "A new state observer for perspective systems," *IEEE Trans. Autom. Control*, vol. 47, no. 4, pp. 658–663, 2002.

[10] L. Ma, C. Cao, N. Hovakimyan, C. Woolsey, and W. E. Dixon, "Fast estimation and range identification in the presence of unknown motion parameters," *IMA J. of App. Math.*, vol. 75, no. 2, pp. 165–189, 2010.

[11] 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, 2008.

[12] 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, 2004.

[13] A. Dani, Z. Kan, N. Fischer, and W. E. Dixon, "Structure and motion estimation of a moving object using a moving camera," in *Proc. Am. Control Conf.*, Baltimore, MD, 2010, pp. 6962–6967.

[14] 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.

[15] D. Liberzon, *Switching in Systems and Control*. Birkhauser, 2003.

[16] M. Sznaier and O. Camps, *Encyclopedia of Complexity and Systems Science*. Springer New York, 2009, ch. Motion Prediction for Continued Autonomy, pp. 5702–5718.

[17] M. Sznaier, O. Camps, N. Ozay, T. Ding, G. Tadmor, and D. Brooks, *Dynamics of Information Systems*. Springer New York, 2010, vol. 40, ch. The Role of Dynamics in Extracting Information Sparsely Encoded in High Dimensional Data Streams, pp. 1–27.

[18] T. Ding, M. Sznaier, and O. Camps, "Receding horizon rank minimization based estimation with applications to visual tracking," in *Proc. IEEE Conf. Decis. Control*, Cancun, Mexico, Dec. 2008, pp. 3446–3451.

[19] M. Ayazoglu, B. Li, C. Dicle, M. Sznaier, and O. I. Camps, "Dynamic subspace-based coordinated multicamera tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 2462–2469.

[20] M. Yang, Y. Wu, and G. Hua, "Context-aware visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 7, pp. 1195–1209, Jul. 2009.

[21] H. Grabner, J. Matas, L. V. Gool, and P. Cattin, "Tracking the invisible: Learning where the object might be," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, Jun. 2010, pp. 1285–1292.

[22] L. Cerman, J. Matas, and V. Hlavac, "Sputnik tracker: Having a companion improves robustness of the tracker," in *Proc. Scand. Conf.*, Oslo, Norway, Jun. 2009, pp. 291–300.

[23] B. R. Fransen, O. I. Camps, and M. Sznaier, "Robust structure from motion and identified dynamics," in *Proc. IEEE. Int. Conf. Comput. Vis.*, Oct. 2005, pp. 772–777.

[24] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, 2006.

[25] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

[26] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 229–241, 2001.

[27] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "A quadratic-complexity observability-constrained unscented Kalman filter for SLAM," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1226–1243, Oct. 2013.

[28] J. Sola, A. Monin, M. Devy, and T. Vidal-Calleja, "Fusing monocular information in multicamera SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 958–968, Oct. 2008.

[29] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Consistency analysis and improvement of vision-aided inertial navigation," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 158–176, Feb. 2014.

[30] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 61–76, Feb. 2012.

[31] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[32] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization

and mapping that provably converges," in *Proc. Int. Joint Conf. Artif. Intell.*, 2003, pp. 1151–1156.
[33] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellised particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.
[34] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 103–123, Jan. 1998.
[35] S. S. Blackman, *Multiple-target Tracking with Radar Applications*. Artech House, 1986.
[36] K. Granstrom and U. Orguner, "A PHD filter for tracking multiple extended targets using random matrices," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5657–5671, Nov. 2012.
[37] K. Wyffels and M. Campbell, "Negative observations for multiple hypothesis tracking of dynamic extended objects," in *Proc. Am. Control Conf.*, 2014.
[38] ——, "Negative information for occlusion reasoning in dynamic extended multiobject tracking," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 425–442, Apr. 2015.
[39] A. Andriyenko, S. Roth, and K. Schindler, "An analytical formulation of global occlusion reasoning for multi-target tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, 2011, pp. 1839–1846.
[40] H. Wei, W. Lu, P. Zhu, S. Ferrari, R. H. Klein, S. Omidshafiei, and J. P. How, "Camera control for learning nonlinear target dynamics via bayesian nonparametric dirichlet-process gaussian-process (dp-gp) models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2014, pp. 95–102.
[41] W. Lu, G. Zhang, S. Ferrari, and I. Palunko, "An information potential approach for tracking and surveilling multiple moving targets using mobile sensor agents," in *Proc. SPIE 8045 Unmanned Syst. Technol. XIII*, 2011.
[42] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy, "A bayesian nonparametric approach to modeling motion patterns," *Auton. Robot.*, vol. 31, no. 4, pp. 383–400, 2011.
[43] M. K. C. Tay and C. Laugier, *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2008, vol. 42, ch. Modelling Smooth Paths Using Gaussian Processes, pp. 381–390.
[44] J. Ko and D. Fox, "Gp-bayesfilters: Bayesian filtering using gaussian process prediction and observation models," *Auton. Robot.*, vol. 27, no. 1, pp. 75–90, 2009.
[45] D. Ellis, E. Sommerlade, and I. Reid, "Modelling pedestrian trajectory patterns with gaussian processes," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2009, pp. 1229–1234.
[46] B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysls for surveillance," *IEEE T. Circ. Syst. Vid.*, vol. 18, no. 8, pp. 1114–1127, Aug. 2008.
[47] T. Campbell, S. Ponda, G. Chowdhary, and J. P. How, "Planning under uncertainty using bayesian nonparametric models," in *AIAA Guid. Navig. Control Conf.*, Aug. 2012.
[48] 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, April 2017.
[49] 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.*, to appear.
[50] P. Ioannou and J. Sun, *Robust Adaptive Control*. Prentice Hall, 1996.
[51] K. Narendra and A. Annaswamy, *Stable Adaptive Systems*. Prentice-Hall, Inc., 1989.
[52] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness*. Upper Saddle River, NJ: Prentice-Hall, 1989.
[53] G. V. Chowdhary and E. N. Johnson, "Theory and flight-test validation of a concurrent-learning adaptive controller," *J. Guid. Control Dynam.*, vol. 34, no. 2, pp. 592–607, Mar. 2011.
[54] G. Chowdhary, "Concurrent learning for convergence in adaptive control without persistency of excitation," Ph.D. dissertation, Georgia Institute of Technology, Dec. 2010.
[55] 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.
[56] H. Schaub and J. Junkins, *Analytical Mechanics of Space Systems*, AIAA, Ed. New York: AIAA Education Series, 2003.
[57] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed., ser. IEE Radar, Sonar, Navigation and Avionics Series. The Institution of Electrical Engineers and The American Institute of Aeronautics and Astronautics, 2004.
[58] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3-D Vision*. Springer, 2004.
[59] B. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *Int. J. Comput. Vision*, vol. 13, no. 3, pp. 331–356, Dec. 1994.
[60] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle, "An analytic solution for the perspective 4-point problem," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 1989, pp. 500–507.
[61] D. DeMenthon and L. S. Davis, "Exact and approximate solutions of the perspective-three-point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 11, pp. 1100–1105, 1992.
[62] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 930–943, 2003.
[63] L. Quan and Z. Lan, "Linear N-point camera pose determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 774–780, 1999.
[64] T. Q. Phong, R. Horaud, A. Yassine, and P. D. Tao, "Object pose from 2-d to 3-d point and line correspondences," *Int. J. Comput. Vision*, vol. 15, no. 3, pp. 225–243, 1995.
[65] D. DeMenthon and L. Davis, "Model-based object pose in 25 lines of code," *Int. J. Comput. Vision*, vol. 15, pp. 123–141, 1995.
[66] N. R. Gans, A. Dani, and W. E. Dixon, "Visual servoing to an arbitrary pose with respect to an object given a single known length," in *Proc. Am. Control Conf.*, Seattle, WA, USA, Jun. 2008, pp. 1261–1267.
[67] K. Dupree, N. R. Gans, W. MacKunis, , and W. E. Dixon, "Euclidean calculation of feature points of a rotating satellite: A daisy chaining approach," *AIAA J. Guid. Control Dyn.*, vol. 31, pp. 954–961, 2008.
[68] M. H. Stone, "The generalized weierstrass approximation theorem," *Math. Mag.*, vol. 21, no. 4, pp. 167–184, 1948.
[69] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and Adaptive Control Design*. New York, NY, USA: John Wiley & Sons, 1995.
[70] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Birkhauser: Boston, 2003.
[71] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.
[72] W. Rudin, *Principles of Mathematical Analysis*. McGraw-Hill, 1976.
[73] S. Garrido-Jurado, R. Munoz-Salinas, F. Madrid-Cuevas, and M. Marin-Jimenez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, Jun. 2014.
[74] "Aruco library," http://www.uco.es/investiga/grupos/ava/node/26.