Target Tracking Subject to Intermittent Measurements Using Attention Deep Neural Networks

Zachary I. Bell, *Member, IEEE*, Runhan Sun, Kyle Volle, Prashant Ganesh, *Member, IEEE*, Scott A. Nivison, *Member, IEEE*, and Warren E. Dixon, *Fellow, IEEE*

***Abstract*—This letter presents a novel estimator and predictor framework for target tracking applications that estimates the pose of a mobile target that intermittently leaves the field-of-view (FOV) of a mobile agent's camera. Specifically, the framework uses an attention deep motion model network (DMMN) to estimate the dynamics of the target when the target is in the agent's FOV and uses the DMMN to predict the position, orientation, and velocity of the target when the target is outside the agent's FOV. A Lyapunov-based stability analysis is performed to determine the maximum dwell-time condition on target measurement availability, and experimental results are provided to demonstrate the performance of the proposed framework.**

***Index Terms*—Adaptive control, deep neural networks, lyapunov-based analysis.**

I. Introduction

MOBILE target tracking tasks typically require mobile agents to use sensors such as cameras that have a limited field-of-view (FOV) which results in intermittent feedback of the target when it leaves the camera's FOV (i.e., the target is not visible to the agent). Intermittent measurements can occur for multiple reasons; for example, when the target is occluded by obstacles or other environmental factors. These obstacles and environmental factors may additionally require a mobile agent to purposely navigate away from a target, causing the target to leave the FOV. These conditions resulting in intermittent measurements which present numerous challenges to estimating the pose and velocity of the target by a mobile agent (cf., [1]–[6]).

Manuscript received 21 March 2022; revised 8 June 2022; accepted 30 June 2022. Date of publication 11 July 2022; date of current version 18 July 2022. This work was supported in part by the Task Order Contract with the Air Force Research Laboratory, Munitions Directorate at Eglin AFB, AFOSR under Award FA9550-18-1-0109, Award FA8651-20-F-1025, and Award FA9550-19-1-0169. Recommended by Senior Editor M. Guay. *(Corresponding author: Zachary I. Bell.)*

Zachary I. Bell and Scott A. Nivison are with the Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL 32542 USA (e-mail: zachary.bell.10@us.af.mil; scott.nivison@us.af.mil).

Runhan Sun, Kyle Volle, Prashant Ganesh, and Warren E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: runhansun@ufl.edu; kvolle@ufl.edu; prashant.ganesh@ufl.edu; wdixon@ufl.edu).

Digital Object Identifier 10.1109/LCSYS.2022.3189949

Intermittency in measurements has traditionally been addressed by using probabilistic estimators. Various types of Kalman filters are widely used approaches that show local convergence of the mean (cf., [7]–[9]). Another approach is to use particle filters which show convergence as the number of samples approaches infinity (cf., [10]–[13]). In contrast, deterministic estimators typically assume boundedness of uncertainties and disturbances in the analysis, yielding uniformly ultimately bounded results (cf., [5] and [6]).

The recent work in [6] presented a novel method to handle intermittent target feedback when using multiple agents to track a single target. In [6], a centralized approach of modeling the motion of the target (i.e., the target's dynamics) is developed using a single layer neural network called the motion model network (MMN) with a fixed basis. The use of an MMN was motivated by the results in [5]; however, while [5] showed convergence over a finite number of cycles of losing and acquiring a target in an agent's FOV using an average dwell-time condition, there was no condition provided to ensure the estimation error didn't exceed the size of the agent's FOV. In practice, reacquiring a target in an agent's FOV can be challenging if the estimation error grows beyond the size of an agent's FOV. In contrast, [6] considered the size of the agent's FOV and developed minimum and maximum dwell-time conditions which dictate the minimum amount of time the target must be in the FOV and the maximum amount of time the target can leave the FOV to ensure the estimation error doesn't exceed the size of the FOV.

The approach in [6] develops dwell-time conditions based on feedback region sizes and was motivated by the results in [14] where dwell-times were determined for path following in feedback-denied environments. In [14], dwell-time conditions were developed to determine how long a mobile agent could remain in a feedback-denied region without the position estimation error growing beyond the size of the feedback region. However, target tracking introduces additional challenges in developing intermittent feedback dwell-time conditions since a target is typically not directly controlled by the mobile agent (i.e., a target is typically not cooperative) and the motion model of the target is unknown. Previous results have considered methods of indirect control of targets through unknown interaction models between targets and agents (cf., [15] and [16]), and developed approaches to learn and influence a target to follow a desired trajectory; however,

a target is generally not guaranteed to be influenced by the relative pose between the target and the agent or that the agent can always follow the target. Following a target while relying on local feedback (i.e., feedback from local sensors such as a camera) to estimate the pose of the target introduces significant challenges over results that assume feedback is always globally available. Furthermore, the majority of these methods use a single layer MMN and only estimate the ideal output layer with a fixed basis which has been shown to reduce performance (cf., [17]).

In this letter, a novel deep MMN (DMMN) is proposed to estimate the motion model of a mobile target that intermittently leaves an agent's FOV and provide dwell-time conditions for the availability of target measurements. Specifically, motivated by [17], the contribution of this letter is the development of a DMMN framework using a deep neural network (DNN) for the basis (i.e., the underlying dynamics features) in contrast to a fixed basis (cf., [6]). Furthermore, recent advances in attention DNNs (cf., [18]–[20]) are incorporated which have shown improvements in image-based pose estimation tasks (cf., [19]).

The DMMN framework is an estimator where the output layer weights are updated online while the target is in the agent's FOV. Simultaneously, motivated by [17], a replay buffer of target poses is collected online while the target is in the agent's FOV. When a sufficient number of target poses are collected, batch updates to the DMMN basis are performed to improve the estimate of the DMMN basis. After a series of batch updates, the prediction of the target's velocity switches to the new basis. Then, when the target leaves the agent's FOV, the DMMN is used as a predictor to estimate the target's pose and velocity while the target remains outside the agent's FOV.

A Lyapunov-based dwell-time analysis is performed to determine the maximum dwell-time condition that ensures the estimation error of the target's pose does not grow beyond the size of the camera's FOV while the target is outside the FOV. While it is not possible to ensure these conditions are satisfied (e.g., since a target is generally not cooperative), this dwell-time condition can be used to determine when the prediction error of the pose has grown too large and gives constraints to subsequently search for the target, improving the chances of finding the target. Experimental results are provided to demonstrate the performance of the proposed DMMN framework using a quadrotor with a downward facing camera tracking a mobile ground vehicle that intermittently leaves the agent's FOV.

II. System Dynamics

As shown in the schematic in Figure 1, three coordinate frames are used to describe the tracking objective inside the tracking environment, denoted as $\mathcal{U} \subset \mathbb{R}^3$, where $\mathcal{U}$ is convex and compact. Since tracking agents are often restricted by environmental factors (e.g., obstacles and battery life), the tracking agent is constrained within a specified operating region denoted as $\mathcal{O}_c \subset \mathcal{U}$, where $\mathcal{O}_c' \triangleq \{p \in \mathcal{U} \mid p \notin \mathcal{O}_c\}$ describes the remaining space outside of $\mathcal{O}_c$ in $\mathcal{U}$. Let the Euclidean space of the target be represented as $\mathcal{M} \subset \mathcal{U}$ (i.e., all the 3D feature points on the target). Feedback of the target's state is only available when the target is in the tracking agent camera's FOV, $\mathcal{M} \subset \mathcal{V}_c$, where $\mathcal{V}_c \subset \mathcal{O}_c$ denotes the

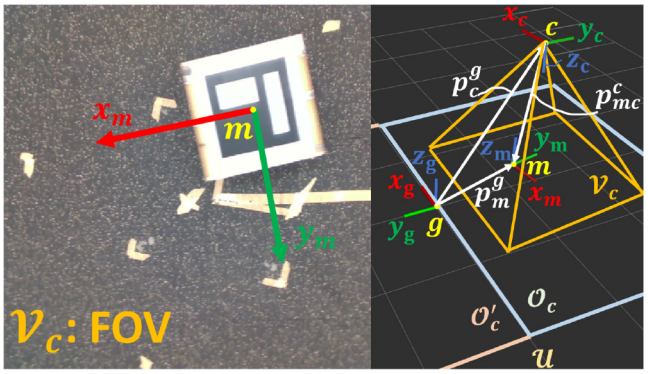

Fig. 1. Kinematic relationship between cooperative and target agents.

Euclidean space contained in the FOV as shown in Figure 1. The tracking agent's camera frame is represented as $\mathcal{F}_c$ and has the origin at the principal point of the camera, denoted as c, with the basis $\{x_c, y_c, z_c\}$, where the z_c axis is along the viewing direction and co-linear with the optical axis, the y_c axis is along the image plane vertical, and the x_c axis is along the image plane horizontal. The frame $\mathcal{F}_g$ denotes the inertial frame with an arbitrarily selected origin g with the basis $\{x_g, y_g, z_g\}$. The mobile target frame is represented as $\mathcal{F}_m$ and has an origin located at an arbitrarily selected feature point on the target, denoted as m, with the basis $\{x_m, y_m, z_m\}$.

A. Target Dynamics

The objective of this letter is to provide a motion model to estimate the pose (i.e., position and orientation) and velocity (i.e., linear and angular velocity) of the target using the tracking agent's camera, despite the target intermittently leaving the FOV. The pose of the tracking agent (i.e., the pose of $\mathcal{F}_c$ with respect to $\mathcal{F}_g$), $\eta_c(t) \in \mathbb{R}^7$, is defined as $\eta_c(t) \triangleq \left[p_c^g(t)^\top \quad q_c(t)^\top\right]^\top$, where $p_c^g(t) \in \mathbb{R}^3$ represents the position of $\mathcal{F}_c$ with respect to $\mathcal{F}_g$ expressed in $\mathcal{F}_g$, and $q_c(t) \in \mathbb{R}^4$ is the quaternion parameterization of $R_c(t) \in \mathbb{R}^{3\times3}$, the rotation matrix representing the orientation of $\mathcal{F}_c$ with respect to $\mathcal{F}_g$. Rotation matrices can be represented using the quaternion parameterization, $q(t) \triangleq [\,q_0(t) \quad q_v^\top(t)\,]^\top \in \mathcal{S}^4$ which has the standard basis $\{1,\ i,\ j,\ k\}$, where $\mathcal{S}^4 \triangleq \{x \in \mathbb{R}^4 | x^\top x = 1\}$, and $q_0(t) \in \mathbb{R}$ and $q_v(t) \in \mathbb{R}^3$ represent the scalar and vector components of $q(t)$, respectively.

Assumption 1: Measurements of the tracking agent's pose, $\eta_c(t)$, are always available using onboard sensors (e.g., an inertial navigation system).

The pose of the target (i.e., the pose of $\mathcal{F}_m$ with respect to $\mathcal{F}_g$), $\eta_m(t) \in \mathbb{R}^7$, is defined as $\eta_m(t) \triangleq \left[p_m^g(t)^\top \quad q_m(t)^\top\right]^\top$, where $p_m^g(t) \in \mathbb{R}^3$ represents the position of $\mathcal{F}_m$ with respect to $\mathcal{F}_g$ expressed in $\mathcal{F}_g$, $q_m(t) \in \mathbb{R}^4$ is the quaternion parameterization of $R_m(t) \in \mathbb{R}^{3\times3}$, the rotation matrix representing the orientation of $\mathcal{F}_m$ with respect to $\mathcal{F}_g$. The pose of the target cannot be directly measured and instead the pose of the camera is used with the measurable relative pose of the target with respect to the camera (i.e., $\mathcal{F}_m$ with respect to $\mathcal{F}_c$ expressed in $\mathcal{F}_c$), $\eta_{mc}(t) \in \mathbb{R}^7$, is defined as $\eta_{mc}(t) \triangleq \left[p_{mc}^c(t)^\top \quad q_{mc}(t)^\top\right]^\top$, where $p_{mc}^c(t) \in \mathbb{R}^3$ represents the position of $\mathcal{F}_m$ with respect to $\mathcal{F}_c$ expressed in $\mathcal{F}_c$, $q_{mc}(t) \in \mathbb{R}^4$ is the quaternion parameterization of $R_{mc}(t) \in \mathbb{R}^{3\times3}$, the rotation matrix representing the orientation of $\mathcal{F}_m$ with respect to $\mathcal{F}_c$. Using

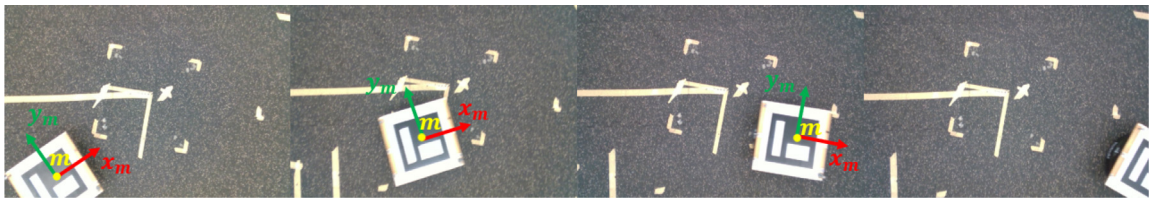

Fig. 2. Generalized schematic of the target tracking objective.

$\eta_c(t)$ and $\eta_{mc}(t)$, the target position is described as illustrated in Figure 1 as

$$p_m^g(t) = p_c^g(t) + R_c(t)p_{mc}^c(t), \tag{1}$$

and the orientation of $\mathcal{F}_m$ with respect to $\mathcal{F}_g$ is

$$q_m(t) = q_c(t) \cdot q_{mc}(t), \tag{2}$$

where $q_c \cdot q_{mc} = \begin{bmatrix} q_{c0} & -q_{cv}^\top \\ q_{cv} & q_{c0}I_3 + q_{cv}^\times \end{bmatrix} q_{mc}$, and $q_{c0}(t) \in \mathbb{R}$ and $q_{cv}(t) \in \mathbb{R}^3$ are the scalar and vector components of $q_c(t)$, respectively.

The target's velocity in the target frame (i.e., velocity of $\mathcal{F}_m$ expressed in $\mathcal{F}_m$), $\varphi_m(t) \in \mathbb{R}^6$, is represented as $\varphi_m(t) \triangleq \begin{bmatrix} v_m(t)^\top & \omega_m(t)^\top \end{bmatrix}^\top$, where $v_m(t), \omega_m(t) \in \mathbb{R}^3$ are the linear and angular velocity of $\mathcal{F}_m$ expressed in $\mathcal{F}_m$, respectively. Using $\varphi_m(t)$, the time derivative of $\eta_m(t)$ yields the velocity of $\mathcal{F}_m$ with respect to $\mathcal{F}_g$ expressed in $\mathcal{F}_g$ as

$$\dot{\eta}_m(t) = f_m(t)\varphi_m(t), \tag{3}$$

where $f_m(t) \triangleq \begin{bmatrix} R_m(t) & 0_{3\times3} \\ 0_{4\times3} & \frac{1}{2}B(q_m(t)) \end{bmatrix} \in \mathbb{R}^{7\times6}$, and $B(q_m(t)) \triangleq \begin{bmatrix} -q_{mv}^\top \\ q_{m0}I_3 + q_{mv}^\times \end{bmatrix} \in \mathbb{R}^{4\times3}$, with the pseudo-inverse property $B(q_m(t))^\top B(q_m(t)) = I_{3\times3}$ and $(\cdot)^\times : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$ represents the skew operator, and $q_{m0}(t) \in \mathbb{R}$ and $q_{mv}(t) \in \mathbb{R}^3$ are the scalar and vector components of $q_m(t)$, respectively (cf., [5]).

Assumption 2: Measurements of $\eta_m(t)$ and $\dot{\eta}_m(t)$ are only available when the target is in the FOV (i.e., $\mathcal{M} \subset \mathcal{V}_c$).

III. Estimation Design

To achieve the objective, a method for estimating the pose of the target agent is developed that uses the target's pose as feedback while the target is in the FOV and predicts the pose and velocity of the target when the target intermittently leaves the FOV. Figure 2 illustrates the target tracking objective where the target enters the tracking agent camera's FOV (left image) and leaves the FOV (right image). To accomplish the tracking objective, an estimator and predictor are developed to estimate the target's pose $\eta_m(t)$. Specifically, a DMMN is used to estimate the target's velocity, $\varphi_m(t)$, when feedback of $\eta_m(t)$ is available (i.e., $\mathcal{M} \subset \mathcal{V}_c$). The DMMN is also used to propagate the pose estimates through time when feedback is unavailable (i.e., $\mathcal{M} \not\subset \mathcal{V}_c$) which occurs when the target is occluded or the target leaves the operating region, $\mathcal{M} \not\subset \mathcal{O}_c$. Let $\rho_c(t) \in \{a, u\}$ be a switching signal indicating if feedback of the target is available (i.e., $\rho_c(t) = a$ when $\mathcal{M} \subset \mathcal{V}_c$) or unavailable (i.e., $\rho_c(t) = u$ when $\mathcal{M} \not\subset \mathcal{V}_c$).

Assumption 3: The pose of the target $\eta_m(t) \in \Omega$ is bounded, where $\Omega \subset \mathbb{R}^7$ is a convex and compact set since $p_m^g(t) \in \mathcal{U}$ and $q_m(t) \in \mathcal{S}^4$ are convex and compact.

Assumption 4: The target's velocity, $\varphi_m(t)$, is described by a locally Lipschitz function of the target's pose, which is not explicitly time dependent. Specifically, $v_m(t) = \varphi_1(\eta_m(t))$ and $\omega_m(t) = \varphi_2(\eta_m(t))$, where $v_m(t)$ and $\omega_m(t)$ are bounded, $\sup_{\eta_m(t)\in\Omega}\{\|v_m(t)\|\} \le \overline{v_m} \in \mathbb{R}_{>0}$ and $\sup_{\eta_m(t)\in\Omega}\{\|\omega_m(t)\|\} \le \overline{\omega_m} \in \mathbb{R}_{>0}$, implying $\varphi_1, \varphi_2 \in \mathbb{R}^3$ are bounded (cf., [5]).

Assumption 4 guarantees there exists a function that can be approximated, using universal function approximators (e.g., neural networks), that describes $v_m(t)$ and $\omega_m(t)$ to an arbitrary level of accuracy via the Stone–Weierstrass Theorem [21]. Furthermore, the Stone–Weierstrass Theorem only ensures the approximation is accurate over a closed interval. Thus, dependence on $\eta_m(t)$ is allowed since it is bounded via Assumption 3. Specifically, from Assumption 4, $\varphi_m(t) = \begin{bmatrix} v_m(t)^\top & \omega_m(t)^\top \end{bmatrix}^\top = \begin{bmatrix} \varphi_1(\eta_m(t))^\top & \varphi_2(\eta_m(t))^\top \end{bmatrix}^\top$ can be approximated using a DNN, that is, the approximation of the DMMN is

$$\varphi_m(t) = W^\top \sigma(\Phi(\eta_m(t))) + \varepsilon(\eta_m(t)), \tag{4}$$

where $W \in \mathbb{R}^{L\times6}$ denotes the constant unknown bounded ideal output layer weight matrix, $\sigma \in \mathbb{R}^L$ denotes the known bounded activation functions corresponding to the output layer, $L \in \mathbb{Z}_{>0}$ denotes the user-defined number of neurons used in the output layer, $\varepsilon \in \mathbb{R}^6$ denotes the unknown bounded function reconstruction error, and $\Phi \in \mathbb{R}^L$ denotes a function that contains the inner layer ideal weights and activation functions of the DNN. Specifically,

$$\Phi(\eta_m(t)) \triangleq (\phi_r \circ \phi_{r-1} \circ \cdots \phi_2 \circ \phi_1)(\eta_m(t)), \tag{5}$$

where $\phi_l = \sigma_l(W_l^\top \phi_{l-1} + b_l)$, $l \in \{1, \ldots, r\}$ with $r \in \mathbb{Z}_{\ge1}$ denoting the user-defined number of inner layers of the DNN, $\phi_0 = \eta_m(t)$ is the input to DNN, $\sigma_l \in \mathbb{R}^{L_l}$ is the activation function for the lth inner layer, $W_l \in \mathbb{R}^{L_{l-1}\times L_l}$ denotes the ideal constant weight matrix for the lth inner layer, and $b_l \in \mathbb{R}^{L_l}$ denotes the ideal constant bias column matrix.

Assumption 5: There exist known constants $\overline{\varphi}_m, \underline{\sigma}, \overline{\sigma} \in \mathbb{R}_{>0}$, where $\sup_{\eta_m(t)\in\Omega}\{\|\varphi_m(\eta_m(t))\|\} \le \overline{\varphi}_m$, $\inf_{\eta_m(t)\in\Omega}\{\sigma(\Phi(\eta_m(t)))\} \ge \underline{\sigma}$, and $\sup_{\eta_m(t)\in\Omega}\{\sigma(\Phi(\eta_m(t)))\} \le \overline{\sigma}$, such that constants $\overline{W}, \overline{\varepsilon} \in \mathbb{R}_{>0}$ can be determined, with $\|W\| \le \overline{W}$ and $\sup_{\eta_m(t)\in\Omega}\|\varepsilon(\eta_m(t))\| \le \overline{\varepsilon}$, (cf., [5], [6], and [17]).

Substituting (4) into (3) yields

$$\dot{\eta}_m(t) = f_m(t)W^\top \sigma(\Phi(\eta_m(t))) + f_m(t)\varepsilon(\eta_m(t)). \tag{6}$$

Let $\widetilde{\eta}_m(t) \in \mathbb{R}^7$ quantify the pose error as

$$\widetilde{\eta}_m(t) \triangleq \eta_m(t) - \widehat{\eta}_m(t), \tag{7}$$

where $\widehat{\eta}_m(t) \in \mathbb{R}^7$ is the estimate of $\eta_m(t)$. Let $\widetilde{W}(t) \in \mathbb{R}^{L\times6}$ quantify the error in the estimate of the ideal output weights

of the DMMN as

$$\widetilde{W}(t) \triangleq W - \widehat{W}(t), \tag{8}$$

where $\widehat{W}(t) \in \mathbb{R}^{L\times 6}$ is the estimate of W.

Since the ideal weights of inner layers of the DNN contained in $\Phi(\eta_m(t))$ are unknown, let $\widehat{\Phi}_k(\eta_m(t)) \in \mathbb{R}^L$, $k \in \{1, 2, \ldots\}$, represent the kth iterative update to approximate $\Phi(\eta_m(t))$. Furthermore, let $T_k \in \mathbb{R}_{\geq 0}$ represent the time when $\widehat{\Phi}_k(\eta_m(t))$ is used to approximate (4), since $\widehat{\Phi}_k(\eta_m(t))$ is updated at a slower timescale using a subsequently defined loss function, (cf., [17]).

A. Target Pose Estimator

While $\rho_c(t) = a$ (i.e., the target is in the FOV and measurements of the target pose, $\eta_m(t)$, are available), an estimator is designed to update the pose estimate; however, a predictor must be used while $\rho_c(t) = u$ (i.e., the target is outside the FOV and measurements of the target pose, $\eta_m(t)$, are unavailable). Let $t_j^a \in \mathbb{R}_{\geq 0}$ represent the jth instance in time when $\rho_c(t) = a$ (i.e., the jth time the target enters the FOV) and let $t_j^u \in \mathbb{R}_{>0}$ represent the jth instance in time when $\rho_c(t) = u$ (i.e., the jth time the target leaves the FOV), where $j \in \{1, 2, \ldots\}$. Since the objective is to track and predict the target's trajectory, dwell-times are defined to quantify the amount of time the target is inside or outside of the FOV. Specifically, let $\Delta t_j^a \triangleq t_j^u - t_j^a$ and $\Delta t_j^u \triangleq t_{j+1}^a - t_j^u$ represent the jth amount of time $\rho_c(t) = a$ and $\rho_c(t) = u$, respectively.

Assumption 6: The target is in the FOV upon initialization (i.e., $\rho_c(0) = a$, $t_1^a = 0$, and $t_1^a < t_1^u$), (cf., [5] and [6]).

Based on the subsequent analysis, the pose estimate update law is designed as

$$\dot{\widehat{\eta}}_m(t) = \begin{cases} f_m(t)\widehat{\varphi}_m(\eta_m(t)) + k_\eta \widetilde{\eta}_m(t) \\ \quad + k_\varepsilon \mathrm{SGN}(\widetilde{\eta}_m(t)), & \rho_c(t) = a, \\ \mathrm{proj}\{\widehat{f}_m(t)\widehat{\varphi}_m(\widehat{\eta}_m(t))\}, & \rho_c(t) = u, \end{cases} \tag{9}$$

where $\mathrm{SGN}(\widetilde{\eta}_m(t)) = \begin{cases} 1, & \widetilde{\eta}_m(t) > 0, \\ 0, & \widetilde{\eta}_m(t) = 0, \\ -1, & \widetilde{\eta}_m(t) < 0, \end{cases}$ is applied element-wise since $\widetilde{\eta}_m$ is a vector, $k_\eta, k_\varepsilon \in \mathbb{R}_{>0}^{7\times 7}$ are constant control gains, $\mathrm{proj}(\cdot)$ is a continuous projection operator defined in [22] with state and velocity bounds which are known under Assumptions 3 and 4, $\widehat{\varphi}_m(\widehat{\eta}_m(t)) \triangleq \widehat{W}^\top(t)\sigma(\widehat{\Phi}_k(\widehat{\eta}_m(t)))$, and $\widehat{f}_m(t) \triangleq \begin{bmatrix} \widehat{R}_m(t) & 0_{3\times 3} \\ 0_{4\times 3} & \frac{1}{2}B(\widehat{q}_m(t)) \end{bmatrix}$.

Remark 1: When the target leaves the FOV, the pose estimate is reset to the last measured pose before predicting the pose using (9) while $\rho_c(t) = u$. This reset reduces the prediction error while $\rho_c(t) = u$ and is only used at the first instances of $\rho_c(t) = u$ (i.e., at $t = t_j^u$, $\widehat{\eta}_m(t_j^u) \mapsto \eta_m(t_j^u)$ implying $\|\widetilde{\eta}_m(t_j^u)\| = 0$).

B. Weight Estimator

The weight estimator update laws are designed based on whether sufficient data has been collected on the trajectory of the target and whether the target is in the FOV.

1) Output Weight Updates: Motivated by the subsequent analysis, the output weight update law $\dot{\widehat{W}}(t)$ is designed

$$\mathrm{vec}\left(\dot{\widehat{W}}(t)\right) = \mathrm{proj}\left(\mu(t), \mathrm{vec}\left(\widehat{W}(t)\right)\right), \tag{10}$$

where $\mathrm{vec}(\cdot)$ is the vectorization operator which stacks $(\cdot)$ column-wise, and $\mu \in \mathbb{R}^{6L}$ is

$$\mu(t) = \begin{cases} \mu_k^a(t), & \rho_c(t) = a, \\ 0_{6L\times 1}, & \rho_c(t) = u, \end{cases} \tag{11}$$

$\mu_k^a(t) \triangleq \mathrm{vec}(\Gamma\sigma(\widehat{\Phi}_k(\eta_m(t)))\widetilde{\eta}_m^\top(t)f_m(t))$, and $\Gamma \in \mathbb{R}_{>0}^{L\times L}$ is a positive definite, constant gain matrix.

2) Inner Weight Updates: The inner weights are updated periodically while $\rho_c(t) = a$ by saving tuples of data to a buffer $\mathcal{B}(t) = \{\eta_m(t_h), \dot{\eta}_m(t_h)\}_{h=1}^{b(t)}$ and when $b(t) > \bar{b}$, an optimization is performed to minimize the mean squared error of the velocity estimates. Specifically, the objective of the $k+1$th batch optimization is to keep $\widehat{W}(t)$ fixed and update the kth approximation of $\Phi(\eta_m(t))$ using the loss,

$$\mathcal{L}_{k+1}(t) = \frac{1}{b(t)}\sum_{h=1}^{b(t)}\left\|\dot{\eta}_m(t_h) - \widehat{W}^\top(t)\sigma\left(\widehat{\Phi}_k(\eta_m(t_h))\right)\right\|^2,$$

where Adam [23] is used to perform the optimization.

IV. Analysis

The subsequent Lyapunov-based analysis provides conditions to ensure the tracking and weight estimation errors remain bounded despite the target intermittently leaving the FOV. The following provides a maximum dwell-time condition (i.e., maximum amount of time the target can leave the FOV) that ensures the target estimation error doesn't grow beyond a user-defined threshold.

Consider a stacked error $\xi(t) \triangleq \left[\widetilde{\eta}_m(t)^\top \quad \mathrm{vec}(\widetilde{W}(t))^\top\right]^\top \in \mathbb{R}^{7+6L}$ and a Lyapunov-based function candidate defined as

$$V(\xi(t)) \triangleq \frac{1}{2}\widetilde{\eta}_m^\top(t)\widetilde{\eta}_m(t) + \frac{1}{2}\mathrm{tr}\left(\widetilde{W}^\top(t)\Gamma^{-1}\widetilde{W}(t)\right), \tag{12}$$

which is bounded as $\underline{\beta_\xi}\|\xi(t)\|^2 \leq V(\xi(t)) \leq \overline{\beta_\xi}\|\xi(t)\|^2$, $\underline{\beta_\xi} \triangleq \frac{1}{2}\min\{1, \lambda_{\min}\{\Gamma^{-1}\}\}$, and $\overline{\beta_\xi} \triangleq \frac{1}{2}\max\{1, \lambda_{\max}\{\Gamma^{-1}\}\}$, where $\lambda_{\min}$ and $\lambda_{\max}$ denote the minimum and maximum eigenvalue of $\{\cdot\}$, respectively.

Theorem 1: The tracking and weight error in $\xi(t)$ is uniformly ultimately bounded while $\rho_c(t) = a$, using the update laws in (9) and (10), in the sense that

$$\|\xi(t)\|^2 \leq \frac{\overline{\beta_\xi}}{\underline{\beta_\xi}}\|\xi\left(t_j^u\right)\|^2 \exp\left(-\beta_a\left(t - t_j^u\right)\right) + \frac{\delta_a}{\beta_a\underline{\beta_\xi}}, \tag{13}$$

where $\beta_a \triangleq \frac{2\lambda_{\min}\{k_\eta\}}{\max\{1, \lambda_{\max}\{\Gamma^{-1}\}\}}$ and $\delta_a \triangleq 4\lambda_{\min}\{k_\eta\}\overline{\varphi_m}^2$.

Proof: Substituting (6), (9)-(11), and the time derivative of (7) and (8) into the time derivative of (12), for $\rho_c(t) = a$, using the bounds on (12), Assumptions 3-5, and simplifying yields

$$\dot{V}(\xi(t)) \leq -\beta_a V(\xi(t)) + \delta_a, \ \rho_c(t) = a. \tag{14}$$

Applying the Comparison Lemma [24, Lemma 3.4] to (14), the bounds on (12), and simplifying yields (13). ■

Theorem 2: The tracking and weight error in $\xi(t)$ is bounded while $\rho_c(t) = u$, using the update laws in (9) and (10), in the sense that

$$\|\xi\left(t_{j+1}^a\right)\|^2 \leq \frac{\overline{\beta_\xi}}{\underline{\beta_\xi}}\|\xi\left(t_j^u\right)\|^2 + \frac{\overline{\Delta t^u \zeta_u}}{\underline{\beta_\xi}}, \tag{15}$$

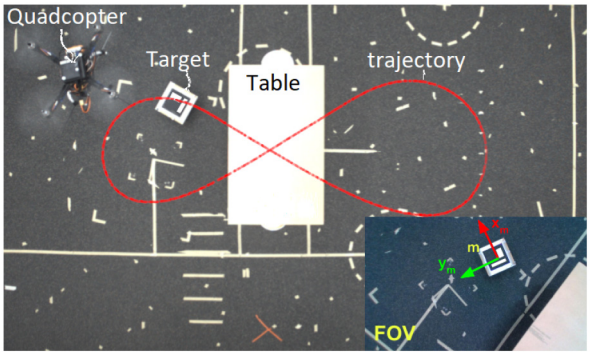

Fig. 3. Overhead view of the quadcopter tracking the ground robot using a camera with the FOV embedded in the bottom right of the image. The target is occluded as it passes under the table in the center. The figure overlays the actual trajectory of the ground robot.

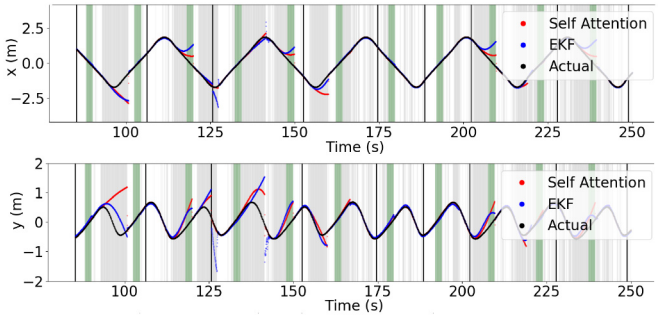

Fig. 4. The estimated and actual x and y positions versus time obtained from the DMMN and the EKF, respectively. Green regions denote occlusions under the table, gray regions denote occlusions from quadcopter drift or image dropout, and the black vertical lines denote the time instances when training occurs.

provided the following maximum dwell-time condition, $\overline{\Delta t^u} \in \mathbb{R}_{>0}$, is satisfied

$$\overline{\Delta t^u} \leq \frac{\overline{\zeta_u}}{4\overline{\varphi}_m^2}, \tag{16}$$

where $\overline{\zeta_u} \in \mathbb{R}_{>0}$ is a user-defined threshold based on the size of the tracking agent's FOV.

Proof: Substituting (6), (9)-(11), and the time derivative of (7) and (8) into the time derivative of (12), for $\rho_c(t) = u$, using the bounds on (12), Assumptions 3-5, and simplifying yields

$$\dot{V}(\xi(t)) \leq \zeta_u(\|\widetilde{\eta}_m(t)\|),\ \rho_c(t) = u, \tag{17}$$

where $\zeta_u(\|\widetilde{\eta}_m(t)\|) \triangleq 2\overline{\varphi}_m\|\widetilde{\eta}_m(t)\|$. For the user-defined threshold on the error to hold, $\zeta_u(\|\widetilde{\eta}_m(t)\|) \leq \overline{\zeta_u} \iff \|\widetilde{\eta}_m(t)\| \leq \zeta_u^{-1}(\overline{\zeta_u})$ resulting in the maximum dwell-time condition in (16). Using the maximum dwell-time constraint in (16), applying the Comparison Lemma [24, Lemma 3.4] to (17), the bounds on (12), and simplifying yields (15). ■

Remark 2: The design of $\overline{\zeta_u}$ can be considered an engineering parameter that is dependent on $\mathcal{O}_c$, $\mathcal{V}_c$, $\eta_{mc}(t)$, and assumptions made about the target (e.g., $\overline{\varphi}_m$).

V. Experiments

The performance of the developed estimation and prediction framework is validated with a quadcopter equipped with a downward facing camera tracking a mobile ground target as

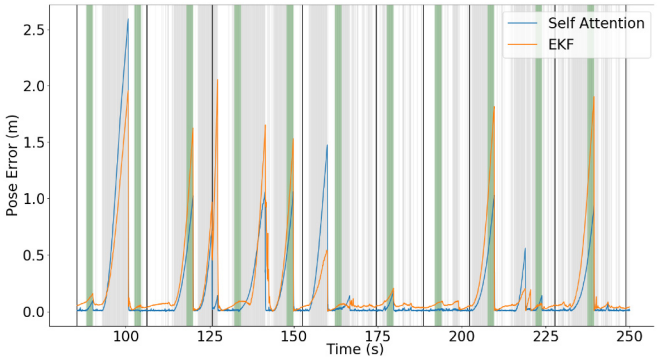

Fig. 5. Pose error in the DMMN and EKF versus time. Green regions denote occlusions under the table, gray regions denote occlusions from quadcopter drift or image dropout, and the black vertical lines denote the time instances when training occurs.

show in Figure 3. Figure 3 shows the mobile target tracks a 4×1 meter figure-8 trajectory using a motion capture system and is equipped with an ArUco marker [25] which is a placeholder for target recognition. The quadcopter is manually piloted using the REEF Estimator [26] for velocity and altitude control and when the ArUco marker is detected in the FOV, the estimator gets the target's pose $\eta_m(t)$ using motion capture; however, as shown in Figure 3, a table occludes the target for approximately 5 seconds every time the target passes under the 1×1 meter portion in the center of the figure-8, shaded green in Figures 4 and 5. Additionally, the grayed regions in Figures 4 and 5 show the target's ArUco marker is frequently not detected because image distortion from camera vibrations in-flight or the target leaves the FOV, where the target remained outside the FOV (i.e., $\rho(t) = u$) for 65% of the experiment. It was assumed the altitude, roll, and pitch of the target were 0, so only the x-y position and yaw of the target were estimated.

The system ran in real-time at 30Hz and the DNN was updated asynchronously 9 times over the experiment. The gains were selected as $k_\eta = 20$ and $\Gamma = I_{10\times 10}$. The DMMN consisted of 3 ReLu layers with 10 basis each, a self-attention block, and a tanh layer at the output with 10 basis. The self-attention block looks for interaction between the features vectors and uses a nonlinear mapping to combine them for prediction. The performance of the DMMN is compared to an extended Kalman filter (EKF). The EKF process standard deviation (SD) was 0.1 m/s and 0.1 rad/s, for linear and angular velocities, respectively. The measurement SD was 0.01 m and 0.02 rad for position and orientation, respectively. The EKF state covariance was initialized to the respective state element values (i.e., pose and velocity state elements were initialized to the measurement and process variance, respectively). Figure 4 shows the individual components of the tracking error over time starting after the first training cycle which occurred at around 90 seconds. Figure 5 shows the norm of the error over that time period. A total of 5 training cycles occurred over the 2 minutes shown in Figures 4 and 5, indicated by the black vertical lines. While feedback was available, pose measurements were stored in a buffer and once 500 measurements were saved, the DMMN was trained as described in Section III-B2 for 75 epochs yielding a loss less than 10^{-4}. Once a training cycle was complete, half the measurements were randomly thrown out and the model was trained after

collecting another 250 new measurements, saving the 250 old measurements for each training cycle. Over the 2.5 minutes of experiment time shown in Figures 4 and 5, the mean squared error of the position was 0.17 meters with the DMMN and 0.23 meters for the EKF. The results showed the DMMN had higher robustness to noise (i.e., pose error after converging was consistently lower), while also performing better overall and in occluded nonlinear regions of the trajectory; however, the EKF outperformed in occluded approximately linear regions since the EKF predicts piece-wise constant velocities. These results motivate research into deep motion model architectures that could improve predictions, which is outside the scope of this result.

For this set of experiments, the user-defined threshold was approximately 1 meter based on the FOV, and based on the maximum linear and angular velocity bounds of 0.5 m/s and 0.5 rad/s, respectively, and the maximum dwell-time was approximately 1 second based on (16) in the analysis; however, feedback of the target was frequently unavailable for longer than 1 second and typically the error didn't exceed the user-defined threshold until after 6 seconds of unavailable feedback implying the dwell-time was conservative. As shown by the gray regions in Figures 4 and 5, the norm of the error was on average approximately 0.1 meters after 1 second of unavailable feedback, 0.2 meters after 2 seconds, 0.5 meters after 4 seconds, and 1.1 meters after 6 seconds. This implies that while the error growth beyond the threshold was exponential as expected, the self-attention DMMN provided better than anticipated performance over periods of unavailable feedback based on the maximum dwell-time. Additionally, the self-attention DMMN performed well overall since the mean squared error of the estimate was 0.59 meters which is within the user-defined threshold of 1 meter.

VI. Conclusion

A novel estimation framework was presented that uses self-attention DNNs to estimate the pose and velocity of a mobile target that intermittently leaves the FOV of a mobile tracking agent equipped with a camera. A Lyapunov-based analysis was used to determine a maximum dwell-time condition on the availability of feedback to determine the maximum amount of time the target could leave the FOV before the estimation error grew beyond a user-defined threshold. The presented experimental results demonstrated that the proposed self-attention DMMN performed better than expected by the Lyapunov-based analysis. Future work will examine methods to improve maximum dwell-time estimates and determine methods of extending this framework to developing consensus DMMNs with cooperative agents.

Acknowledgment

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.

References

[1] S. Avidan and A. Shashua, "Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 348–357, Apr. 2000.

[2] J. Y. Kaminski and M. Teicher, "A general framework for trajectory triangulation," *J. Math. Imag. Vis.*, vol. 21, no. 1, pp. 27–41, 2004.

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

[4] D. Chwa, A. P. Dani, and W. E. Dixon, "Range and motion estimation of a monocular camera using static and moving objects," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 4, pp. 1174–1183, Jul. 2016.

[5] 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.

[6] C. G. Harris, Z. I. Bell, R. Sun, E. A. Doucette, J. W. Curtis, and W. E. Dixon, "Target tracking in the presence of intermittent measurements by a network of mobile cameras," in *Proc. IEEE Conf. Decis. Control*, 2020, pp. 5962–5967.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.

[12] M. J. McCourt, J.-P. Ramirez-Paredes, E. A. Doucette, and J. W. Curtis, "A hybrid estimation algorithm for tracking an adversarial team," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 2017, pp. 2273–2278.

[13] J.-P. Ramirez-Paredes, E. A. Doucette, J. W. Curtis, and N. R. Gans, "Distributed information-based guidance of multiple mobile sensors for urban target search," *Auton. Robots*, vol. 42, pp. 375–389, Feb. 2018.

[14] 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.

[15] R. A. Licitra, Z. I. Bell, and W. E. Dixon, "Single-agent indirect herding of multiple targets with uncertain dynamics," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 847–860, Aug. 2019.

[16] P. Deptula, Z. I. Bell, F. M. Zegers, R. A. Licitra, and W. E. Dixon, "Approximate optimal influence over an agent through an uncertain interaction dynamic," *Automatica*, vol. 134, pp. 1–13, Dec. 2021.

[17] R. Sun, M. L. Greene, D. M. Le, Z. I. Bell, G. Chowdhary, and W. E. Dixon, "Lyapunov-based real-time and iterative adjustment of deep neural networks," *IEEE Control Syst. Lett.*, vol. 6, pp. 193–198, 2021.

[18] A. Vaswani *et al.*, "Attention is all you need," in *Advanced in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2017, pp. 5998–6008.

[19] B. Wang, C. Chen, C. X. Lu, P. Zhao, N. Trigoni, and A. Markham, "Atloc: Attention guided camera localization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 10393–10401.

[20] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[21] M. H. Stone, "The generalized Weierstrass approximation theorem," *Math. Mag.*, vol. 21, no. 4, pp. 167–184, 1948.

[22] Z. Cai, M. S. de Queiroz, and D. M. Dawson, "A sufficiently smooth projection operator," *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 135–139, Jan. 2006.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

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

[25] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image Vis. Comput.*, vol. 76, pp. 38–47, Aug. 2018.

[26] J. H. Ramos, P. Ganesh, W. Warke, K. Volle, and K. Brink, "REEF estimator: A simplified open source estimator and controller for multi-rotors," in *Proc. IEEE Nat. Aerosp. Electron. Conf. (NAECON)*, 2019, pp. 606–613.