The Observability in Unobservable Systems

Wei Kang[1], Liang Xu[2] and Hong Zhou[3]

Abstract— In this paper, we introduce the concept of observability of selected state variables for systems that may not be fully observable. For their estimation, we introduce and exemplify a deep filter, which is a neural network specifically designed for the estimation of selected state variables without computing the trajectory of the entire system. The observability definition is quantitative rather than a yes or no answer so that one can compare the level of observability between different sensor locations.

I. INTRODUCTION

In this paper, we study the observability and estimation methods of selected state variables for systems that are not necessarily observable. In control theory, the observability is a widely used concept that defines the feasibility of estimating the state, or initial state, of dynamical systems based on their outputs. In this paper, we address the problem of partial observability. If a system is not observable, is it possible to take the advantage of its output to estimate a part of the state variables? Even if a system is observable, is that possible to estimate only the selected state variables without using a full scale filter that estimates all variables in the state space? The answer to these questions is important for various reasons. The ever increasing complexity and dimension of dynamical systems in science and engineering result in many problems for which the system is either unobservable or computationally too expensive to provide a real-time estimation of the entire system states. Some examples of such systems include numerical weather prediction and large swarms of unmanned vehicles. In [10] and [14], for instance, deep learning is applied to nowcasting, i.e., predicting the future rainfall intensity in a local region over a relatively short period of time without running a full-scale data assimilation system. In [2], it is proved that partially estimating some parameters in swarms of unmanned vehicles is possible when the overall system is unobservable.

There is a huge literature on the theory of nonlinear estimation and filtering. It is worth to note that deep learning as a tool of nonlinear filtering, namely deep filter, is analyzed in [15]. We adopt a similar approach in Section IV. Different from the problem addressed in [15] where the system is observable, we use deep filters to approximate a selected individual variable in a system that may not be fully observable. Measuring observability quantitatively based on the

observabililty Gramian is essential to the type of problems studied in this paper. For related literature, the interested readers are referred to [1], [2], [4], [5], [6], [7], [8], [9], [11], [12], [13] and references therein.

In this paper, we introduce the concept of observability of selected state variables. For their estimation, we introduce and exemplify a deep filter, which is a neural network specifically designed for the estimation of selected state variables without computing the trajectory of the entire system.

II. OBSERVABILITY

Consider a discrete-time system

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{f}(\mathbf{x}(k)), \\ \mathbf{y}(k) &= \mathbf{h}(\mathbf{x}(k)),\end{aligned} \tag{1}$$

where $k \geq 0$ is an integer and

$$\begin{aligned}\mathbf{x} &= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^n \\ \mathbf{y} &= \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^m\end{aligned} \tag{2}$$

are the state and output variables, respectively. Given $\mathbf{y}(0), \mathbf{y}(1), \cdots, \mathbf{y}(K)$, where $K \geq 1$ is an integer, we define the observability of an individual state. Without loss of generality, let us consider the observability of $x_1(K)$. It is worth to note that we measure the observability of the final state at $k = K$. However, similar ideas in this section can be applied to the observability of the initial state $x_1(0)$. In general, the quantitative measure of observability has different value for the initial and final states. Following the general definition of observability introduced in [5], we define the observability of $x_1(K)$. Consider a trajectory, $\{\mathbf{x}(k); 0 \leq k \leq K\}$. We want to define the observability around this trajectory. Let $\{\hat{\mathbf{x}}(k); 0 \leq k \leq K\}$ denote a trajectory close to $\{\mathbf{x}(k)\}$. Define ρ using the following constrained maximization,

$$\rho^2 = \max_{\hat{\mathbf{x}}(0), \cdots, \hat{\mathbf{x}}(K)} \{(\hat{x}_1(K) - x_1(K))^2\}$$

subject to

$$\begin{aligned}\hat{\mathbf{x}}(k+1) &= \mathbf{f}(\hat{\mathbf{x}}(k)), & k &= 0, 1, \cdots, K-1 \\ \hat{\mathbf{y}}(k) &= \mathbf{h}(\hat{\mathbf{x}}(k)), & k &= 0, 1, \cdots, K \\ \sum_{k=0}^{K} \|\hat{\mathbf{y}}(k) - \mathbf{y}(k)\|_2^2 &\leq \epsilon^2\end{aligned} \tag{3}$$

where $\epsilon > 0$ is a constant representing the upper bound of output variations and $0 \leq \rho \leq \infty$ is called the ambiguity in the estimation of $x_1(K)$. The ratio ρ/ϵ is called the *unobservability index*, a term adopted from [9]. One can use a different norm to define ρ in (3). However, adopting the 2-norm makes it easy to approximate the unobservability index using an observability Gramian. If the output variable

[1]Wei Kang, Department of Applied Mathematics, Naval Postgraduate School, Monterey, California & University of California, Santa Cruz, California, USA wkang@nps.edu

[2] Liang Xu, Marine Meteorology Division, Naval Research Laboratory, Monterey, California, USA liang.xu@nrlmry.navy.mil

[3] Hong Zhou, Department of Applied Mathematics, Naval Postgraduate School , Monterey, California, USA hzhou@nps.edu

represents the sensor information, then it has measurement error. The true value of $\mathbf{y}(k)$ is unknown. What we know is that the truth should be in a neighborhood of the measured data. Therefore, we allow in (3) the output to take any value in an ϵ-neighborhood. The trajectory $\{\hat{\mathbf{x}}(k); 0 \le k \le K\}$ in (3) that has the largest variation of $|\hat{x}_1(K) - x_1(K)|$ is considered the worst possible estimation of $x_1(K)$ whose output is within an ϵ-neighborhood of the measured $\mathbf{y}$. To summarize, if the value of the unobservability index, ρ/ϵ, is small, then $x_1(K)$ is observable. More specifically, among all approximate trajectories satisfying $\sum_{k=0}^{K} \|\hat{\mathbf{y}}(k) - \mathbf{y}(k)\|_2^2 \le \epsilon^2$, the worse estimation error of $x_1(0)$ is ρ. The range of the unobservability index value for observable systems depends on the scale of the uncertainties in $\mathbf{y}$. In this paper, we use this quantitative measure of observability to find individual variables that can be estimated accurately and to compare the observability between different sensor locations.

Consider a linear system,

$$\begin{aligned} \mathbf{x}(k+1) &= A\mathbf{x}(k); \\ \mathbf{y}(k) &= H\mathbf{x}(k) \end{aligned} \quad (4)$$

where $A \in \mathbb{R}^{n \times n}$ and $H \in \mathbb{R}^{m \times n}$. Then (3) is equivalent to

$$\begin{aligned} \rho^2 = \max_{\Delta \mathbf{x}_0} \ & \Delta \mathbf{x}_0^\intercal F^\intercal F \Delta \mathbf{x}_0, \\ \text{subject to} \quad & \\ & \Delta \mathbf{x}_0^\intercal G \Delta \mathbf{x}_0 = \epsilon^2, \end{aligned} \quad (5)$$

where $\Delta \mathbf{x}_0$ represents $\hat{\mathbf{x}}(0) - \mathbf{x}(0)$ and

$$\begin{aligned} G &= \sum_{k=0}^{K} (A^\intercal)^k H^\intercal H A^k, \\ F &= \begin{bmatrix} 1 & 0 \cdots 0 \end{bmatrix} A^K. \end{aligned} \quad (6)$$

The computation of the unobservability index boils down to numerically solving the constrained quadratic maximization (5)-(6). The matrix G is, in fact, the observability Gramian, a symmetric matrix that is used to measure observability in control theory [4], [5]. For problems that have high dimensions, numerically solving (5) is not straightforward if the overall system is weakly observable or unobservable because, in this case, the condition number of G can be very large. Techniques of computational linear algebra have to be applied when solving (5). Due to length limitations, these techniques will be addressed in another paper.

For a nonlinear system, the computation is more difficult because (3) is a nonconvex problem, which is not easy to solve. On the other hand, we can approximate the observability by solving (5) based on the linearization of the nonlinear system along nominal trajectories. Specifically, G and F can be computed empirically using the central difference method. Let $\delta > 0$ be a small number. Around a trajectory $\{\mathbf{x}(k); 0 \le k \le K\}$, we compute $2n$ trajectories

$$\begin{aligned} \mathbf{x}^{\pm i}(0) &= \mathbf{x}(0) \pm \delta \mathbf{e}_i, \\ \mathbf{x}^{\pm i}(k+1) &= \mathbf{f}(\mathbf{x}^{\pm i}(k)), \quad k = 0, 1, \cdots, K-1 \\ \mathbf{y}^{\pm i}(k) &= \mathbf{h}(\mathbf{x}^{\pm i}(k)), \qquad k = 0, 1, \cdots, K \end{aligned} \quad (7)$$

for $i = 1, 2, \cdots, n$, where $\mathbf{e}_i$ is the ith unit vector in $\mathbb{R}^n$. Define

$$\begin{aligned} \Delta^i \mathbf{y}(k) &= (\mathbf{y}^{+i}(k) - \mathbf{y}^{-i}(k))/(2\delta), \\ \Delta \mathbf{y}(k) &= \begin{bmatrix} \Delta^1 \mathbf{y}(k) & \Delta^2 \mathbf{y}(k) & \cdots & \Delta^n \mathbf{y}(k) \end{bmatrix}, \quad (8) \\ \Delta^i x_1(K) &= (x_1^{+i}(K) - x_1^{-i}(K))/(2\delta). \end{aligned}$$

Then, G and F are approximated by

$$\begin{aligned} G &= \sum_{k=0}^{K} \Delta \mathbf{y}(k)^\intercal \Delta \mathbf{y}(k), \\ F &= \begin{bmatrix} \Delta^1 x_1(K) & \Delta^2 x_1(K) & \cdots & \Delta^n x_1(K) \end{bmatrix} \end{aligned} \quad (9)$$

In the following, the value of ρ/ϵ is approximated by solving (5) using the empirical approximation of G and F in (9). This approach avoids the requirement of solving the nonconvex optimization problem (3).

III. ESTIMATING STATE VARIABLES OF UNOBSERVABLE SYSTEMS

In this section, we use an example to illustrate the idea of estimating a selected state variable when the overall system is unobservable. Consider Burgers' equation

$$\begin{aligned} & \frac{\partial U(x,t)}{\partial t} + U(x,t) \frac{\partial U(x,t)}{\partial x} = \kappa \frac{\partial^2 U(x,t)}{\partial x^2}, \\ & U(x,0) = U_0(x), \\ & U(0,t) = 0, \ U(L,t) = 0. \end{aligned} \quad (10)$$

where $(x,t) \in [0,L] \times [0,T]$. The solution is approximated by solving a finite dimensional discretized system. The discretization is based on central difference in space and 4th order Runge-Kutta in time (see, for instance, [7]). Let $N_t > 0$ and $N_x > 0$ be integers. The discretized trajectory is represented as follows,

$$\mathbf{u}(k) = \begin{bmatrix} u_1(k), u_2(k), \cdots, u_{N_x-1}(k) \end{bmatrix}^\intercal, \quad (11)$$

for $k = 0, 1, \cdots, N_t$. In (11), $u_i(k)$ represents $U(x,t)$ evaluated at the grid point (x_i, t_k), where

$$\begin{aligned} x_i &= i\Delta x, & i &= 1, 2, \cdots, N_x - 1 \\ t_k &= k\Delta t, & k &= 0, 1, \cdots, N_t, \\ \Delta x &= L/N_x, \\ \Delta t &= T/N_t. \end{aligned} \quad (12)$$

Note that the boundaries $u_0(k) = 0$ and $u_{N_x}(k) = 0$ are known. In this example, we set $T = 5$, $L = 2\pi$, $\kappa = 0.14$, $N_x = 50$ and $N_t = 100$. For the output, we assume that sensors are located at x_{20}, x_{21}, x_{29} and x_{30}, i.e.,

$$\mathbf{y}(k) = \begin{bmatrix} u_{20}(k) & u_{21}(k) & u_{29}(k) & u_{30}(k) \end{bmatrix}^\intercal \quad (13)$$

For nonlinear systems, the observability depends on the location of the trajectory. For analysis, we take a data-driven approach. We randomly select $5,000$ initial states around $U(x,0) = 0$. Their empirical Gramians are computed using (7)-(9) for $K+1 = 10$, i.e. measurements of 10 time steps of $\mathbf{y}$ are used in the state estimation. The smallest eigenvalue of G at the samples is shown in Figure 1. The eigenvalues are small, all around 10^{-10}. This implies that the observability
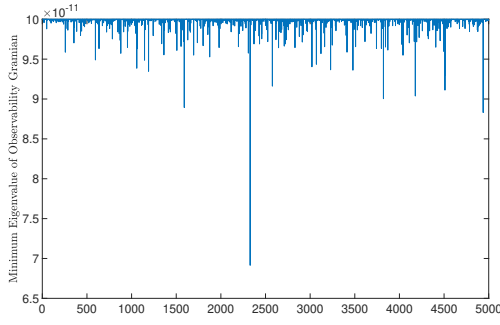
Fig. 1. The minimum eigenvalue of the observability Gramian at $5,000$ sample points.

of the system is extremely weak, or the system is practically unobservable.

Unscented Kalman filter (UKF) [3] is applied to the system. Because the observability is extremely weak, we do not expect an accurate estimation of all state variables. In the simulation, we apply i.i.d. random noise to $\mathbf{y}$. The standard deviation is $\sigma = 0.028$. Due to the extremely weak observability, the filter cannot correct the initial estimation error. One example is shown in Figure 2. The initial error in the estimation of u_{12} cannot be corrected efficiently.

Fig. 2. The true value of $u_{12}(k)$ and its UKF estimation. The state $u_{12}(K)$ is practically unobservable because $\rho/\epsilon = 1.1884 \times 10^4$.

Although the overall system is unobservable, some individual state variables can still be reasonably observable under the same output function (13). For example, by solving (5) using the empirical Gramian (9), the unobservability index of $u_{25}(K)$ is $\rho/\epsilon = 4.3702$ if $K + 1 = 10$ and $\rho/\epsilon = 0.7940$ if $K + 1 = 20$. This is in sharp contrast to the unobservable state $u_{12}(K)$ shown in Figure 2, for which $\rho/\epsilon = 1.1884 \times 10^4$. What does this mean? Assume that the sensor error of each measurement is 0.028. The total l_2 norm of sensor error for 4×10 measurements ($K + 1 = 10$) is about $\|e\|_2 = 0.1771$ and $\|e\|_2 = 0.2504$ for 4×20 measurements ($K + 1 = 20$). Multiplying $\|e\|_2$ by the value of ρ/ϵ, it implies that the worst error in the estimation of $u_{25}(K)$ is about 0.7740 when $K + 1 = 10$ and 0.1988 when $K + 1 = 20$. They may not seem to be very small. But we would like to emphasize that they are the worst possible error. An estimator, such as UKF, that optimizes the estimation avoids the worst scenario.

In fact, the UKF estimation of $u_{25}(k)$ has a much smaller error than the worst case. Shown in Figures 3-4 , the UKF estimation of $u_{25}(k)$ converges to the true value quickly; the initial error is reduced by more than 90% after $k = 20$ steps.
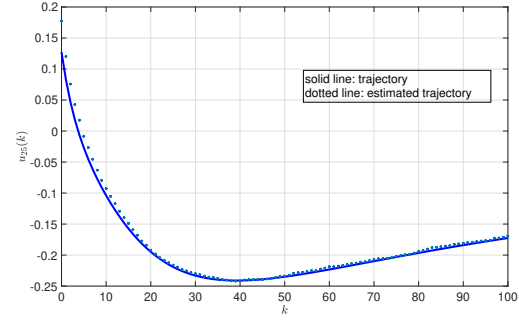

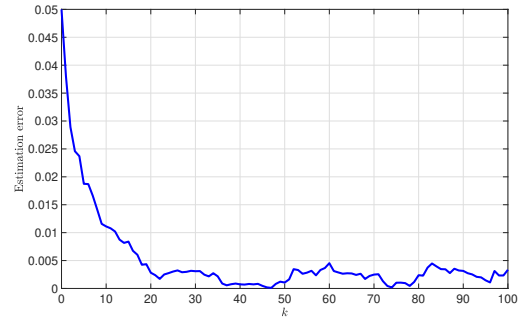Fig. 3. The true value of $u_{25}(k)$ in a trajectory and its UKF estimation.

Fig. 4. The error of UKF estimation $|\hat{u}_{25}(k) - u_{25}(k)|$.

IV. DEEP FILTER

A UKF filter estimates all state variables even if the goal is to find an estimation of a single selected variable. This is inefficient. In fact, for some high dimensional systems, a real-time estimation of the entire system is simply impossible due to the high computational load. In this section, we introduce a deep filter. The method is based on deep learning. An advantage of a deep filter is that the computation is focused on the selected state variable to be estimated, without the requirement of running a full scale filter that estimates all state variables simultaneously. In [15], deep neural networks are applied to approximate state variables based on the output. In this section, we apply a similar idea. But different from [15] where the system is observable, we use deep neural networks to approximate an individual variable in a system that may not be fully observable.

In this study, a feedforward neural network (Figure 5) is a scalar-valued function, $\mathbf{z} \in \mathbb{R}^p \to u^{NN} \in \mathbb{R}$

$$u^{NN}(\mathbf{z}) = g_M \circ g_{M-1} \circ \cdots g_1(\mathbf{z}) \tag{14}$$

where $g_k(\bar{\mathbf{z}}) = \sigma(\mathbf{W}_k \bar{\mathbf{z}} + \mathbf{b}_k)$, $\bar{\mathbf{z}}$ is a vector (may have different dimensions in different layers of the network), σ is a vector-valued activation function such as the hyperbolic

317

tangent, the logistic or ReLU function. The input dimension, p, depends on the dimension of $\mathbf{y}$ and the number of measurements used in estimation. For example, if $\mathbf{y} \in \mathbb{R}^m$ and if the estimation is based on the information $\{\mathbf{y}(k); 0 \leq k \leq K\}$, then $p = m(K+1)$. The value of u^{NN} is an approximation of an individual state variable of the system. In the training process, we minimize the following loss function

$$l(\mathbf{W}, \mathbf{b}) = \frac{1}{|\mathcal{S}_{training}|} \sum_{(\mathbf{z}, u(\mathbf{z})) \in \mathcal{S}_{training}} (u^{NN}(\mathbf{z}) - u(\mathbf{z}))^2,$$
(15)

where

$$\mathcal{S}_{training} = \left\{ (\mathbf{z}, u(\mathbf{z})); \begin{array}{l} \mathbf{z} \text{ is in a set of random} \\ \text{points in state space} \end{array} \right\}$$
(16)

The training is based on the BFGS algorithm to find the parameters $\mathbf{W}$ and $\mathbf{b}$ that minimize $l(\mathbf{W}, \mathbf{b})$. For validation, the accuracy of $u^{NN}(\mathbf{z})$ is evaluated using another data set, $\mathcal{S}_{validation}$.
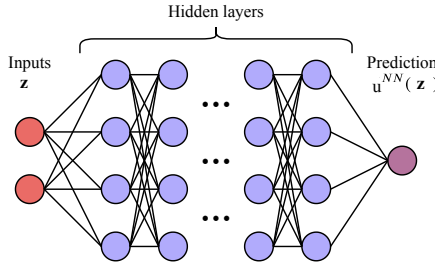

Fig. 5. A feedforward neural network

Let us consider the discretized Burgers' equation in Section III as an illustrative example. We solve the equation using random initial conditions to generate data. Then a neural network is trained so that it can estimate $u_{25}(K)$ based on the sequence of outputs $\{\mathbf{y}(k); 0 \leq k \leq K\}$. The data sets, $\mathcal{S}_{training}$ and $\mathcal{S}_{validation}$, are generated using trajectories from random initial states in the following form

$$\mathbf{u}_i(0) = \sum_{j=0}^{N_F} \left(\alpha_j \cos\left(\frac{2\pi j}{L} x_i\right) + \beta_j \sin\left(\frac{2\pi j}{L} x_i\right) \right)$$
$$i = 1, 2, \cdots, N_x - 1$$
(17)

subjecting to $\sum \alpha_j = 0$ to satisfy the boundary condition (we simply set $\alpha_{N_F} = -\sum_{j < N_F} \alpha_j$). The parameters $\alpha_j \sim N(0, \sigma)$ and $\beta_j \sim N(0, \sigma)$ are i.i.d. random variables. In this example, $N_F = 3$, $\sigma = 0.3$ and $K + 1 = 10$. For each data point in $\mathcal{S}_{training}$ and $\mathcal{S}_{validation}$, $\mathbf{z}$ is the vector in $\mathbb{R}^{m(K+1)}$ obtained by reshaping $\{\mathbf{y}(k); 0 \leq k \leq K\}$. For each random initial condition and its trajectory, three data points are generated based on $\{\mathbf{y}(s+k); 0 \leq k \leq K\}$, where $s = 0$ and random integers s_1 and s_2. For each data point, $u(\mathbf{z})$ is assigned the corresponding value of $u_{25}(s+K)$. The size of the data sets is

$$|\mathcal{S}_{training}| = |\mathcal{S}_{validation}| = 3 \times 10^4.$$
(18)

In the validation, the error of u_{25}^{NN} is measured by the root-mean-square error (RMSE)

$$\sqrt{\sum_{(\mathbf{z}, u(\mathbf{z})) \in \mathcal{S}_{validation}} \frac{(u^{NN}(\mathbf{z}) - u(\mathbf{z}))^2}{|\mathcal{S}_{validation}|}}$$
(19)

Two sets of sensor locations are used,

$$\text{Case 1}: \mathbf{y}(k) = \begin{bmatrix} u_{20}(k) & u_{21}(k) & u_{29}(k) & u_{30}(k) \end{bmatrix}^\mathsf{T}$$
$$\text{Case 2}: \mathbf{y}(k) = \begin{bmatrix} u_{18}(k) & u_{19}(k) & u_{30}(k) & u_{31}(k) \end{bmatrix}^\mathsf{T}.$$
(20)

Using the first set of sensor locations (Case 1), the averaged observability index of u_{25} over the validation data set is $\rho/\epsilon = 4.05$. For Case 2, the number is 43.18, which indicates that the observability is weaker than Case 1. The trained neural networks have eight layers and 32 neurons in each layer. The activation function is the hyperbolic tangent. An i.i.d. Gaussian noise is added to $\mathbf{y}$. The standard deviation is 0.028, which is about 5% of the averaged range of $\{u_{25}(k); 0 \leq k \leq K\}$ in the data. The RMSE of u^{NN} is shown in Table I. When there is no sensor noise in $\mathbf{y}$, the accuracy of both cases are similar. However, in the presence of sensor noise, the RMSE in Case 1, that has a higher observability, is significantly smaller than Case 2. A sample trajectory is shown in Figures 6 and 7 for Case 1 and Case 2, respectively. The neural network approximation using sensor locations in Case 2 (dotted line in Figure 7) has larger error than Case 1 in Figure 6 when $k \leq 30$, the transition phase when the trajectory approaches an equilibrium. The trained neural network, or the deep filter, provides the approximated trajectory of $u_{25}(k)$ without the need of computing other state variables in $\mathbb{R}^{49}$.

TABLE I
RMSE OF u_{25}^{NN}

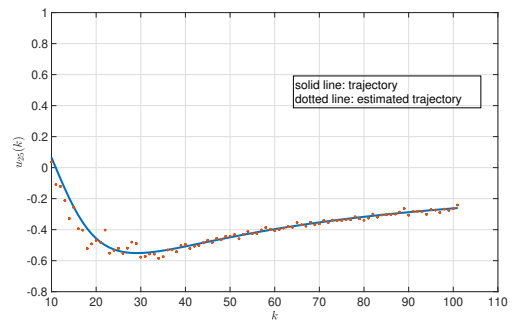|	ρ/ϵ		RMSE
Sensor location		noise free	0.0056
Case 1	4.05	with noise	0.0225
Sensor location		noise free	0.0066
Case 2	43.18	with noise	0.0801

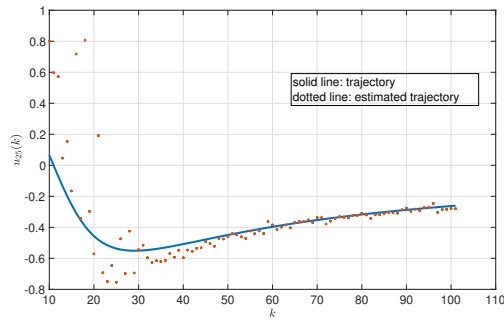Fig. 6. A trajectory and its neural network estimation - Case 1

Fig. 7. A trajectory and its neural network estimation - Case 2

V. CONCLUSIONS

In control theory, the definition of observability based on the observability matrix rank condition is a yes or no answer rather than a quantitative measure. In addition, this definition is about the observability of the overall system state, not applicable to individual state variables. For high dimensional problems, the classical definition of observability does not serve the purpose when estimating the entire system trajectory is either impossible or unnecessary. In this paper, we introduce a quantitative measure of observability for selected state variables. For their estimation, we introduce and exemplify a deep filter, which is a neural network specifically designed for the estimation of selected state variables without computing the trajectory of the entire system. From the examples, both estimations from UKF and deep filter agree with the measure of observability, i.e., the sensor locations that have higher observability (or lower value of the unobservability index) result in more accurate estimation. For future research, more questions are raised than what answered in this paper such as more testing examples that have higher dimensions, finding efficient numerical methods for the quadratic maximization problem that defines the unobservability index, improving the training process for the deep filter, and applying the idea to numerical weather prediction including nowcasting of selected local areas.

ACKNOWLEDGMENT AND DISCLAIMER

REFERENCES

[1] T. Breiten and K. Kunisch, Neural network based nonlinear observer, arXiv:2003.07269v1, March, 2020.
[2] Q. Gong, W. Kang, C. Walton, I. Kaminer and H. Park, Partial observability analysis of an adversarial swarm model, J. Guidance, Control, and Dynamics, Vol. 43(2), pp 250-261, 2020.
[3] S. Julier, J. Uhlmann and H. F. Durrant-Whyte, A new method for the nonlinear transformation of means and covariances in filters and estimators. IEEE Trans. Automat. Contr. 45, 477–482, 2000.
[4] T. Kailath, Linear Systems, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1980.
[5] W. Kang and L. Xu, Computational analysis of control systems using dynamic optimization, arXiv:0906.0215v2, July, 2009.
[6] W. Kang and L. Xu, "A quantitative measure of observability and controllability," Proceedings of the 48th IEEE Conference on Decision and Control, IEEE, New York, pp. 6413–6418, 2009.
[7] W. Kang and L. Xu, Optimal placement of mobile sensors for data assimilations, Tellus A, 64, 17133, 2012.
[8] W. Kang and L. Xu, Partial observability for some distributed parameter systems, International Journal of Dynamics and Control, 2014, online (DOI) 10.1007/s40435-014-0087-4, Vol. 2(4) pp 587-596, 2014.
[9] A. J. Krener and K. Ide, Measure of unobservability, Proc. IEEE Conference on Decision and Control and Chinese Control Conference, Shanghai, China, December, 2009.
[10] S. Otsuka, M. Ohhigashi, V. P. Huynh, P. Tandeo, and T. Miyoshi, A deep-learning approach to three-dimensional precipitation nowcasting, The 2nd R-CCS International Symposium, Kobe, Japan, February 17, 2020.
[11] N. Powel, and K. Morgansen, "Empirical observability Gramian rank condition for weak observability of nonlinear systems with control," IEEE 54th Annual Conference on Decision and Control, IEEE,York, pp. 6342–6348, 2015.
[12] J. Qi, K. Sun, and W. Kang, "Optimal PMU placement for power system dynamic state estimation by using empirical observability Gramian," IEEE Transactions on Power Systems, Vol. 30, No. 4, pp. 2041–2054, 2015.
[13] J. Qi, K. Sun, and W. Kang, "Adaptive optimal PU placement based on empirical observability Gramian," IFAC-PapersOnLine, Vol. 49, No. 18, pp. 482–487, 2016.
[14] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-C. Woo, Convolutional LSTM network: a machine learning approach for precipitation nowcasting, arXiv:1506.04214_v2, 2015.
[15] Q. Zhang, G. Yin and L. Y. Wang, Deep filtering, arXiv:2008.03878v1, August, 2020.