Performance Bounds for Neural Network Estimators: Applications in Fault Detection

Navid Hashemi, Mahyar Fazlyab, Justin Ruths

***Abstract*— We exploit recent results in quantifying the robustness of neural networks to input variations to construct and tune a model-based anomaly detector, where the data-driven estimator model is provided by an autoregressive neural network. In tuning, we specifically provide upper bounds on the rate of false alarms expected under normal operation. To accomplish this, we provide a theory extension to allow for the propagation of multiple confidence ellipsoids through a neural network. The ellipsoid that bounds the output of the neural network under the input variation informs the sensitivity - and thus the threshold tuning - of the detector. We demonstrate this approach on a linear and nonlinear dynamical system.**

I. Introduction

The rise in interest in data-driven techniques is a response to the need for better models of complex systems. Model-based fault and anomaly detection in dynamical systems typically employs first-principle models (Newton's and Kirchhoff's Laws, reaction kinetics, etc.) of systems to identify discrepancies between the observed and predicted sensor measurements. However, to use this general approach on large-scale operations, such as petro-chemical refineries and power distribution grids, operators need scalable and efficient techniques for creating models. Data-driven tools offer a compelling option because models can be devised simply from past data, do not require intimate knowledge of system parameters, and can be recomputed regularly to update the dynamics that may change over time.

Artificial neural networks, in particular, provide relatively easy training and generalization, simple architecture, good ability to approximate nonlinear functions, and robust to inexact input data [1]. Neural networks can be used to identify and control nonlinear dynamic systems because they can approximate a wide range of nonlinear functions. Over the past two decades, fault detection has employed neural networks in a variety of ways, including using them as classifiers to categorize normal or various faulty behaviors [2] and using them as a model of the system to provide an estimate for subsequent fault detection [3], [4]. In the latter case, neural network estimators can be implemented using an iterative approach (in which the past estimate(s) is part of the input to produce the next estimate) or an autoregressive approach (in which a finite history of past measurements is used to produce the next estimate) [5]-[7]. The detection problem becomes particularly challenging when the fault is unknown or cannot be modeled (e.g., we want to build a detector that is sensitive to potentially unknown faults or anomalies, like attacks). In this context, tuning of the detector is important to balance the sensitivity of the detector with the prevalence of false alarms and while data-driven techniques are not new to fault detection, tuning to-date has been done empirically. In this paper we leverage recent results to quantify the robustness of neural networks [8], [9] to develop a novel data-driven anomaly detector with guarantees on the upper limit of the false alarm rate.

Quantifying the robustness of neural networks is originally motivated by their vulnerability to adversarial attacks, i.e., carefully chosen small input perturbations that can drastically change their output. To this end, a plethora of tools have been developed to bound the output of neural networks for a given range of inputs (e.g., the set of plausible attacks) [9]-[15]. Our particular interest in this paper is the uncertainty propagation technique put forth in [8], in which the authors develop a semi-definite program that propagates an input ellipsoid (e.g., confidence region of a density function) through the neural network to obtain *guaranteed* ellipsoidal over-approximation of the output set. In another context, Monte Carlo sampling, the Unscented Transform (UT) and Extended Kalman Filtering (EKF) have been used to take a set of samples from the input distribution, propagate them through the neural network, and approximate the first and second moments of the output distribution from them [16], [17]. These sampling methods scale to larger neural networks but they lack formal guarantees.

II. Background

Consider a discrete-time dynamical system whose state update can be described by an *unknown* continuous function, $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^n$,

$$x_{k+1} = \mathcal{F}(x_k), \tag{1}$$

which may arise as a freely evolving dynamical system or as a feedback control system. Our observations (sensor measurements) of the system are linear combinations of the states corrupted by additive zero-mean noise $v_k \in \mathbb{R}^p$, with known covariance Σ_v,

$$y_k = Hx_k + v_k, \tag{2}$$

where the sensor matrix $H \in \mathbb{R}^{p\times n}$ is known. Because the system model is unknown, or possibly too complicated with too many parameters to use in an effective way, we use a data-driven estimator with the assumption that it is possible to construct an accurate nonlinear auto-regressive (NARX)

N. Hashemi and J. Ruths are with the Department of Mechanical Engineering, The University of Texas at Dallas. Email: (nxh150030, jruths)@utdallas.edu. M. Fazlyab is with the John Hopkins Mathematical Institute for Data Sciences. Email: mahyarfazlyab@jhu.edu

model of the output observations based on past measurement values (i.e., the system is fully observable),

$$y_{k+1} = f(\mathcal{Y}_{k,N}), \tag{3}$$

where $\mathcal{Y}_{k,N} = \textbf{vec}\left[\{y_{k-i}\}_{i=0}^{N}\right] \in \mathbb{R}^{p(N+1)}$.

Again, the mapping $f : \mathbb{R}^{n\times p} \to \mathbb{R}^p$ is *unknown* or otherwise too complicated to form from first-principle models, so we approximate this mapping through supervised training of a feedforward neural network from time-series recordings of sensor measurements. At each time $k \geq N$ the network is provided the vector of current and N past measurements $\mathcal{Y}_{k,N}$ as input training data and the next measurement y_{k+1} as the labeled output data of the neural network. The network is trained under a large number of varying initial conditions.

The ℓ-layer neural network, trained with input data $\mathcal{Y}_{k,N}$ and labeled data y_{k+1}, is

$$\begin{aligned} \zeta_k &= [\ \mathcal{Y}_{k,N}\ ]^\top, \\ z^0 &= \zeta_k, \\ z^{t+1} &= \phi(W^t z^t + b^t), \quad t = 0, 1, \ldots, \ell - 1, \\ \hat{y}_{k+1} &= W^\ell z^\ell + b^\ell, \end{aligned} \tag{4}$$

with ReLU activation function $\psi(s_i) = \max(0, s_i), \forall s_i \in \mathbb{R}$, and $\phi([s_1, s_2, .., s_d]) = [\psi(s_1), \psi(s_2), ..., \psi(s_d)]$. We train the model to compute optimal weight matrices and bias vectors and we call the output of the neural network, $\hat{y}_{k+1}$, as the estimated/predicted output.

The primary role of the training process is to refine the neural network to produce *accurate* estimates, i.e., predictions $\hat{y}_k$ that are close to the actual measurements y_k. The purpose of this paper is not to refine the training process to improve accuracy, but to quantify the robustness of the prediction to input uncertainties for a *given* trained neural network (e.g., uncertainty caused by sensor noise). Further, we aim to use the robustness bounds as a way to quantify normal behavior from abnormal behavior.

Remark 1: The NARX architecture is one of several ways to construct a neural network estimator of a dynamical system. An alternative choice is to feed the past estimation $\hat{y}_k$ as an input to the prediction for $\hat{y}_{k+1}$. The autoregressive approach we use here simplifies the training process and will also simplify the robustness quantification since the bounds on the neural network inputs are constant across time.

III. Neural Network-Based Anomaly Detector

In the same spirit as propagating the estimation covariance of a Luenberger observer (e.g., Kalman Filter), evaluating the robustness of a neural network estimator identifies the quality of the estimation - quantifying how much the estimator can reduce the influence of the noise on the prediction. At the same time, quantifying the inherent variation in the predictions due to uncertainty (noise) also provides a bound on the variation observed during normal behavior.

Consider the noisy measurement y_k from (2) at a particular time step k, which is composed of the non-noisy (deterministic) "ideal" measurement,

$$y_k^* = Hx_k, \tag{5}$$

and a sample of the sensor noise, v_k. If a different realization of the sensor noise v_k was instead used, our goal is to bound how different the prediction $\hat{y}_{k+1}$ would be under these two sensor realization scenarios. Said differently, we would like to assess the robustness of the prediction to the inherent perturbations caused by sensor noise.

Because the support of the noise could be large, or possibly infinite (e.g., Gaussian noise), it is practically useful to truncate the noise distribution at a desired confidence set, such that the probability $\bar{p}$ that a noise sample is drawn from within the confidence set is a desired value (typically close to one). An ellipsoidal confidence set can be constructed using a scaled version of the covariance of the noise distribution as the shape matrix and characterized by

$$\Pr[v_k^\top \Sigma_v^{-1} v_k \leq \alpha] = \Pr[v_k^\top \underbrace{(\alpha\Sigma_v)^{-1}}_{\bar{\Sigma}_v} v_k \leq 1] = \bar{p}. \tag{6}$$

This defines the $\bar{p}$-confidence ellipsoid on the sensor noise, $\mathcal{E}(0, \bar{\Sigma}_v)$, with an ellipsoid with center μ and shape matrix Σ defined as

$$\mathcal{E}(\mu, \Sigma) := \left\{\xi \mid (\xi - \mu)^\top \Sigma^{-1} (\xi - \mu) \leq 1\right\}, \tag{7}$$

and the size of the ellipsoid to match the desired confidence level $\bar{p}$ is chosen by selecting $\alpha = 2\Gamma^{-1}(\frac{p}{2}, \bar{p})$, using the inverse regularized lower incomplete gamma function and p is the number of measurements at each time step [18].

One way to interpret this bound is that since $y_k = y_k^* + v_k$, the actual measurement y_k is contained within an ellipsoid $\mathcal{E}(y_k^*, \bar{\Sigma}_v)$, i.e., centered at the ideal measurement y_k^* (see Fig. 1a). However, we can also see that $y_k^* = y_k - v_k$ which says the opposite - that the ideal measurement is contained within the ellipsoid centered at the actual measurement, $y_k^* \in \mathcal{E}(y_k, \bar{\Sigma}_v)$ (see Fig. 1b). The latter interpretation is practically useful because we have access to the actual measurements, but not the ideal (noise-free) measurements.

The primary contribution of this paper is to construct and tune a model-based anomaly detector that employs a neural network to produce an estimate. Using Fig. 1 as a guide, this objective can be decomposed into two pieces

1) the framework to quantify the robustness of the neural network under input perturbations (N blue input confidence ellipsoids), $\mathcal{E}(y_{k-N}, \bar{\Sigma}_v), \cdots, \mathcal{E}(y_k, \bar{\Sigma}_v)$ to produce an ellipsoidal bound on the predictions (orange ellipsoid) which we will denote by $\mathcal{E}(\hat{\mu}_{k+1}, \hat{\Sigma}_{k+1})$; and
2) the definition of a detector that uses this (orange) ellipsoid to bound normal behavior through the use of the geometric sum.

We address the detector definition first in the following subsection before presenting the mathematical framework to create the prediction bound in Section IV.

A. Detector Definition

Given the ellipsoidal confidence sets that quantify the perturbation of the inputs to the neural network estimator, in Section IV we will compute an ellipsoidal bound on the output of the estimate, which we call the prediction

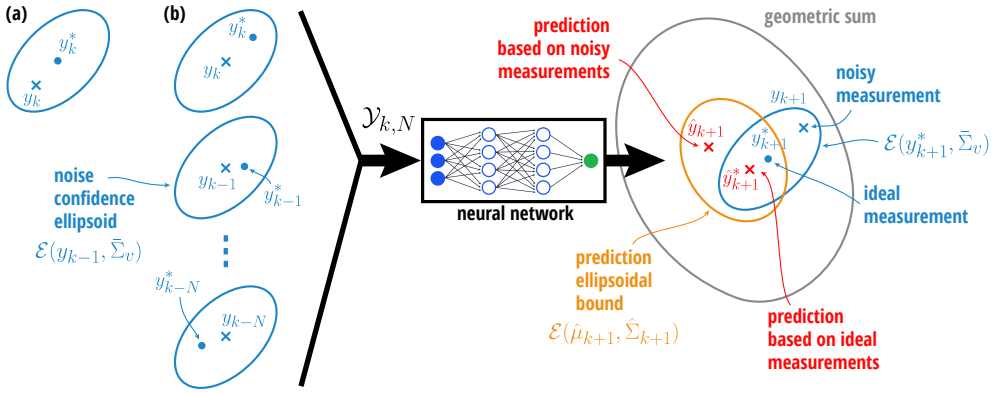


Fig. 1: The proposed detector relies on the development of a prediction ellipsoid (orange) $\mathcal{E}(\hat{\mu}_{k+1}, \hat{\Sigma}_{k+1})$ that bounds the variation of the estimate due to the noise inherent in the input measurements y_k, y_{k-1}, …, y_{k-N} and captured by the input ellipsoids $\mathcal{E}(y_{k-i}, \bar{\Sigma}_v)$, $i = 0, \ldots, N$. The framework to compute the prediction ellipsoid is presented in Section IV and the detector that uses this ellipsoid to decide between normal and anomalous behavior is presented in Section III.

bound $\mathcal{E}(\hat{\mu}_{k+1}, \hat{\Sigma}_{k+1})$. This bound guarantees that any $N+1$ past measurement vectors which stay within their respective confidence sets are mapped by the neural network to a point within the prediction bound ellipsoid.

From Fig. 1, since the ideal past measurements $y^*_{k-N}, \ldots, y^*_k$ are within the input confidence ellipsoids, we know the neural network output estimate $\hat{y}^*_{k+1}$ will be located within the prediction bound. The accuracy of the estimator can be interpreted as the distance between this prediction based on ideal measurements $\hat{y}^*_{k+1}$ and the next ideal measurement y^*_{k+1}. In our framework presented here, we will assume that the trained model is accurate such that this distance is relatively small and can be neglected; however, an ellipsoidal bound on the estimation accuracy could be easily integrated into our detector definition.

As discussed above, the actual noisy measurement at $k+1$ is, with probability $\bar{p}$ within the confidence ellipsoid centered at the ideal measurement, $y_{k+1} \in \mathcal{E}(y^*_{k+1}, \bar{\Sigma}_v)$. Although we do not know the ideal measurement, we know it is within the prediction bound. This leads us to suggest a detector that evaluates normal behavior as measurements based on the geometric sum between the prediction bound and the confidence ellipsoid.

Definition 1: A model-based anomaly detector that employs an autoregressive neural network to produce an estimate from past measurements raises alarms using the following logic:

$$\begin{cases} y_{k+1} \in \mathcal{E}(\hat{\mu}_{k+1}, \hat{\Sigma}_{k+1}) \oplus \mathcal{E}(0, \bar{\Sigma}_v) & \rightarrow \quad \text{no alarm}, \\ \text{otherwise} & \rightarrow \quad \text{alarm}, \end{cases} \tag{8}$$

where the prediction bound is characterized by a center $\hat{\mu}_{k+1}$ and shape matrix $\hat{\Sigma}_{k+1}$; the noise $\bar{p}$-confidence ellipsoid is characterized by the shape matrix $\bar{\Sigma}_v$; and $\oplus$ denotes the geometric (Minkowski) sum of two sets $\mathcal{S}_1 \oplus \mathcal{S}_2 = \{s_1 + s_2 \mid s_1 \in \mathcal{S}_1,\ s_2 \in \mathcal{S}_2\}$.

Remark 2: Since the geometric sum of two ellipsoids is, in general, not an ellipsoid, it is best to verify this inclusion directly rather than to first approximate the sum as an ellipsoid. There are a number of computational techniques to verify the inclusion of a point in the geometric sum of ellipsoidal sets such as linear matrix inequalities [19] or geometric methods to compute the exact geometric sum [20].

The purpose of a detector is to alert the operator of a system to potential anomalies, such as faults or attacks. The number of *false alarms*, i.e., alarms raised during normal operation, is a key way to assess the performance of a detector. Tuning the detector to balance sensitivity and false alarms is a key step towards making a detector usable in practice.

Proposition 1: The detector defined in Definition 1 has a false alarm rate that is upper bounded by $1 - \bar{p}^{N+2}$.

Proof: The fact that the sensor noise is independent allows us to quantify probabilities easily. The confidence ellipsoids are used $N+1$ times for times $k-N, \ldots, k-1, k$ (the input ellipsoids) and once for time $k+1$ in the geometric sum. Thus since v_k is independent, the probability that all these $v_{k-N}, \ldots, v_{k+1}$ fall within their confidence sets is $\bar{p}^{N+2}$. Because the prediction ellipsoid is an outer bound and because there could be noise realizations that lie outside their confidence ellipsoids but still remain inside the prediction bound, this probability is a lower bound. Thus under normal operation, we expect the probability of generating a false alarm to be at most $1 - \bar{p}^{N+2}$. ■

IV. Robustness of Prediction

Our detector design leverages the ability to compute a bound on the predictions made by the neural network estimator under perturbations to the input. To accomplish this bound, we use the formulation described in [8], which provides an ellipsoidal bound on the output of a neural network given a single ellipsoidally bounded input vector. Here, as seen in Fig. 1, our approach uses multiple ellipsoidally bounded input vectors and, therefore, requires an extension to the method in [8]. The fundamental difference in these

$$E_i = \left[\begin{array}{ccc|c|c} 0_{n_i\times(\sum_{j=1}^{i-1} n_j)} & I_{n_i} & 0_{n_i\times(\sum_{j=i+1}^{q} n_j)} & 0_{n_i\times(\sum_{t=2}^{\ell+1} N_j)} & 0_{n_i\times 1} \\ \hline 0_{1\times n_\Gamma} & & & 0_{1\times(\sum_{t=2}^{\ell+1} N_j)} & 1 \end{array}\right] \qquad (\star)$$

two approaches in the context of our problem is at what stage the confidence level truncation is applied. In our proposed multi-ellipsoid input approach we construct a confidence ellipsoid for each individual measurement. To use the single ellipsoid input approach of [8] directly, we would define an overall confidence ellipsoidal set on the entire stacked input vector. The multiple ellipsoid approach has a few key benefits: (1) while the noise is independent, the center of the input ellipsoids $y_{k-N}, \ldots, y_k$ are correlated due to the deterministic dynamics of the system, which complicates concatenating the measurements; (2) the input ellipsoids all have the same shape matrix whereas the single input ellipsoid shape matrix would be time dependent; and (3) in Appendix I we show that the multiple ellipsoid input approach results in a less conservative ellipsoidal prediction bound.

Here we follow the general approach in [8], [9], introducing the updates needed for multiple ellipsoidally bounded inputs. Consider the following general neural network with ℓ hidden layers,

$$\begin{aligned} \Gamma_k &= [\gamma_{1,k}^\top, \ \gamma_{2,k}^\top, \ ..., \gamma_{q,k}{}^\top]^\top \\ z^0 &= \Gamma_k \\ z^{t+1} &= \phi(W^t z^t + b^t), \quad t = 0, 1, ..., \ell - 1 \\ \pi(z^0) &= W^\ell z^\ell + b^\ell, \end{aligned} \qquad (9)$$

with $\gamma_{i,k} \in \mathbb{R}^{n_i}$, $\Gamma_k \in \mathbb{R}^{n_\Gamma}$ ($\sum_{j=1}^q n_i = n_\Gamma$), $\pi(z^0) \in \mathbb{R}^{n_\pi}$ and $z^t \in \mathbb{R}^{N_t}$. By concatenating all the post-activation values as $\mathbf{z} = \left[\ z^0{}^\top, \ z^1{}^\top, ..., z^\ell{}^\top\right]^\top \in \mathbb{R}^{N_\mathbf{z}}$ ($N_\mathbf{z} = \sum_{t=0}^{\ell} N_t$), and defining the "selector matrices" S^t such that $z^t = S^t\mathbf{z}$, we can rewrite the network as

$$\begin{aligned} \Gamma_k &= [\gamma_{1,k}^\top, \ \gamma_{2,k}^\top, \ ..., \gamma_{q,k}{}^\top]^\top \\ z^0 &= S^0\mathbf{z} = \Gamma_k \\ B\mathbf{z} &= \phi(A\mathbf{z} + b) \\ \pi(z^0) &= W^\ell S^\ell \mathbf{z} + b^\ell, \end{aligned} \qquad (10)$$

where

$$A = \begin{bmatrix} W^0 & 0 & \ldots & 0 & 0 \\ 0 & W^1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & W^{\ell-1} & 0 \end{bmatrix}, \quad b = \begin{bmatrix} b^0 \\ b^1 \\ \vdots \\ b^{\ell-1} \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & I_{n_1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & I_{n_\ell} \end{bmatrix}. \qquad (11)$$

Suppose each input $\gamma_{i,k}$ takes values inside the ellipsoid $\mathcal{E}(\mu_{\gamma_{i,k}}, \Sigma_{i,k})$. Our goal is to bound the resulting output $\pi(z^0)$ of the neural network by an ellipsoid. To this end, we first abstract the neural network via Quadratic Constraints [9] and then use the S-procedure to propagate the q input ellipsoids through the network. We begin with the following definition.

Definition 2 (Neural Network Abstraction by QCs [9]): : Let $\phi : \mathbb{R}^d \to \mathbb{R}^d$ and suppose $\mathcal{Q} \subset \mathbb{S}^{2d+1}$ is the set of all symmetric and indefinite matrices Q such that the inequality

$$\begin{bmatrix} x \\ \phi(x) \\ 1 \end{bmatrix}^\top Q \begin{bmatrix} x \\ \phi(x) \\ 1 \end{bmatrix} \geq 0, \quad \forall\ x \in \mathcal{X}, \qquad (12)$$

holds for all $x \in \mathbb{R}^d$. Then we say ϕ satisfies the quadratic constraint defined by $\mathcal{Q}$.

Lemma 1: Consider the neural network described in (10).

1) Suppose $\gamma_{i,k} \in \mathcal{E}(\mu_{\gamma_{i,k}}, \Sigma_{i,k})$. Then for any $\tau_i \geq 0$ we have
$$\begin{bmatrix} \mathbf{z} \\ 1 \end{bmatrix}^\top \left(\sum_{i=1}^{q} \tau_i M_i\right) \begin{bmatrix} \mathbf{z} \\ 1 \end{bmatrix} \geq 0, \qquad (13)$$
where
$$M_i = E_i^\top \begin{bmatrix} -\Sigma_{\gamma_{i,k}}^{-1} & \Sigma_{\gamma_{i,k}}^{-1}\mu_{\gamma_{i,k}} \\ \mu_{\gamma_{i,k}}^\top \Sigma_{\gamma_{i,k}}^{-1} & -\mu_{\gamma_{i,k}}^\top \Sigma_{\gamma_{i,k}}^{-1}\mu_{\gamma_{i,k}} + 1 \end{bmatrix} E_i,$$
and E_i is defined as in $(\star)$.
2) Let $U \in \mathbb{S}^{n_\pi}$, $V \in \mathbb{R}^{n_\pi}$. Suppose $\mathbf{z}$ satisfies the the quadratic inequality
$$\begin{bmatrix} \mathbf{z} \\ 1 \end{bmatrix}^\top M_{out} \begin{bmatrix} \mathbf{z} \\ 1 \end{bmatrix} \leq 0, \qquad (14)$$
where
$$M_{out} = \begin{bmatrix} W^\ell S^\ell & b^\ell \\ 0 & 1 \end{bmatrix}^\top \begin{bmatrix} U^2 & UV \\ V^\top U & V^\top V - 1 \end{bmatrix} \begin{bmatrix} W^\ell S^\ell & b^\ell \\ 0 & 1 \end{bmatrix}.$$
Then we have $\pi(z^0) \in \mathcal{E}(-U^{-1}V, U^{-2})$.

Proof: (1) The proof is a slight modification of [8]. We know that each input $\gamma_{i,k}$ is bounded by the ellipsoid $\mathcal{E}(\mu_{\gamma_{i,k}}, \Sigma_{\gamma_{i,k}})$, which means

$$(\gamma_{i,k} - \mu_{\gamma_{i,k}})^\top \Sigma_{\gamma_{i,k}}^{-1} (\gamma_{i,k} - \mu_{\gamma_{i,k}}) < 1. \qquad (15)$$

This can be rewritten as

$$\begin{bmatrix} \gamma_{i,k} \\ 1 \end{bmatrix}^\top \begin{bmatrix} -\Sigma_{\gamma_{i,k}}^{-1} & \Sigma_{\gamma_{i,k}}^{-1}\mu_{\gamma_{i,k}} \\ \mu_{\gamma_{i,k}}^\top \Sigma_{\gamma_{i,k}}^{-1} & -\mu_{\gamma_{i,k}}^\top \Sigma_{\gamma_{i,k}}^{-1}\mu_{\gamma_{i,k}} + 1 \end{bmatrix} \begin{bmatrix} \gamma_{i,k} \\ 1 \end{bmatrix} \geq 0.$$

Using the selector matrix E_i defined in $(\star)$, we can rewrite the preceding inequality as

$$\begin{bmatrix} \mathbf{z} \\ 1 \end{bmatrix}^\top \underbrace{E_i^\top \begin{bmatrix} -\Sigma_{\gamma_{i,k}}^{-1} & \Sigma_{\gamma_{i,k}}^{-1}\mu_{\gamma_{i,k}} \\ \mu_{\gamma_{i,k}}^\top \Sigma_{\gamma_{i,k}}^{-1} & -\mu_{\gamma_{i,k}}^\top \Sigma_{\gamma_{i,k}}^{-1}\mu_{\gamma_{i,k}} + 1 \end{bmatrix} E_i}_{M_i} \begin{bmatrix} \mathbf{z} \\ 1 \end{bmatrix} \geq 0,$$

This implies for any $\tau_i \geq 0$, the inequality (13) holds. (2) For the proof of the second part, we rewrite $\pi(z^0) \in \mathcal{E}(-U^{-1}V,\ U^{-2})$ as

$$\begin{bmatrix}\pi(z^0)\\ 1\end{bmatrix}^\top \begin{bmatrix}U^2 & UV\\ V^\top U & V^\top V-1\end{bmatrix}\begin{bmatrix}\pi(z^0)\\ 1\end{bmatrix} \leq 0. \tag{16}$$

This can be mapped to the space of vector $\mathbf{z}$ as,

$$\begin{bmatrix}\mathbf{z}\\ 1\end{bmatrix}^\top \begin{bmatrix}W^\ell S^\ell & b^\ell\\ 0 & 1\end{bmatrix}^\top \begin{bmatrix}U^2 & UV\\ V^\top U & V^\top V-1\end{bmatrix}\begin{bmatrix}W^\ell S^\ell & b^\ell\\ 0 & 1\end{bmatrix}\begin{bmatrix}\mathbf{z}\\ 1\end{bmatrix} \leq 0,$$

which characterizes M_{out} provided in the lemma. ■

The quadratic constraint for the activation layers, denoted by $\mathcal{Q}$ is presented in [9] (Lemma 4) and remains unchanged in our formulation. Now consider the implicit equation $B\mathbf{z} = \phi(A\mathbf{z}+b)$ in (10) describing the neural network, where ϕ satisfies the quadratic constraint defined by $\mathcal{Q}$. By Definition 2 this implies that for any $Q \in \mathcal{Q}$,

$$\begin{bmatrix}\mathbf{z}\\ 1\end{bmatrix}^\top \underbrace{\begin{bmatrix}A & b\\ B & 0\\ 0 & 1\end{bmatrix}^\top Q \begin{bmatrix}A & b\\ B & 0\\ 0 & 1\end{bmatrix}}_{M_{mid}(Q)}\begin{bmatrix}\mathbf{z}\\ 1\end{bmatrix} \geq 0, \tag{17}$$

Returning to our main goal, bounding the output $\pi(z^0)$ by an ellipsoid given ellipsoidal bounds on the inputs $\gamma_{i,k}$, we know that the quadratic inequalities (13) and (17) hold and we would like to conclude the quadratic inequality (14). To this end we use the S-procedure. Therefore, the multiple ellipsoidally bounded input version of Theorem 1 (Output covering ellipsoid) of [8] can be presented as follows.

Theorem 1: Consider the multi-layer neural network described by (9). Suppose $\gamma_i \in \mathcal{E}(\mu_{\gamma_{i,k}}, \Sigma_{\gamma_{i,k}})$ and ReLU activation function ϕ satisfies the quadratic constraints defined by (17). Let $U \in \mathbb{S}^{n_\pi}$, $V \in \mathbb{R}^{n_\pi}$ be two matrices that satisfy $M(\tau, Q, U, V) \preceq 0$ for some $Q \in \mathcal{Q}$ and $\tau \in \mathbb{R}^q_+$, where

$$M = \left[\begin{array}{ccc|c} & & & 0_{(N_\mathbf{z}-N_\ell)\times n_\pi}\\ \multicolumn{3}{c|}{\sum_{i=1}^q \tau_i M_i + M_{mid}(Q) - ee^\top} & W^{\ell\top}U\\ & & & b^{\ell\top}U + V^\top\\ \hline 0_{n_\pi\times(N_\mathbf{z}-N_\ell)} & UW^\ell & Ub^\ell+V & -I_{n_\pi}\end{array}\right]. \tag{18}$$

Then we have $\pi(z^0) \in \mathcal{E}_{\pi(z^0)} = \mathcal{E}(-U^{-1}V, U^{-2})$.

Proof: According to the first part of Lemma 1, in order to guarantee $\pi(z^0) \in \mathcal{E}_{\pi(z^0)} = \mathcal{E}(-U^{-1}V, U^{-2})$, we need to provide a sufficient condition to satisfy the inequality,

$$\begin{bmatrix}\mathbf{z}\\ 1\end{bmatrix}^\top M_{out}\begin{bmatrix}\mathbf{z}\\ 1\end{bmatrix} \leq 0. \tag{19}$$

Suppose the following matrix inequality holds,

$$\sum_{i=1}^q \tau_i M_i + M_{mid}(Q) + M_{out} \preceq 0, \tag{20}$$

for some $\tau_i \geq 0$, $Q \in \mathcal{Q}$. Based on (20) we can conclude,

$$\sum_{i=1}^q \underbrace{\begin{bmatrix}\mathbf{z}\\ 1\end{bmatrix}^\top \tau_i M_i \begin{bmatrix}\mathbf{z}\\ 1\end{bmatrix}}_{\geq 0} + \underbrace{\begin{bmatrix}\mathbf{z}\\ 1\end{bmatrix}^\top M_{mid}\begin{bmatrix}\mathbf{z}\\ 1\end{bmatrix}}_{\geq 0} + \begin{bmatrix}\mathbf{z}\\ 1\end{bmatrix}^\top M_{out}\begin{bmatrix}\mathbf{z}\\ 1\end{bmatrix} \leq 0, \tag{21}$$

which provides what we need from M_{out}. The matrix inequality in (20) is not linear in U and V. However, it can be made linear through the Schur complement. First, rewrite M_{out} as

$$\begin{aligned} M_{out} = &\begin{bmatrix}W^\ell S^\ell & b^\ell\\ 0 & 1\end{bmatrix}^\top \begin{bmatrix}U & V\end{bmatrix}^\top \begin{bmatrix}U & V\end{bmatrix}\begin{bmatrix}W^\ell S^\ell & b^\ell\\ 0 & 1\end{bmatrix} -\\ &\begin{bmatrix}W^\ell S^\ell & b^\ell\\ 0 & 1\end{bmatrix}^\top \begin{bmatrix}0_{1\times n_\pi} & 1\end{bmatrix}^\top \begin{bmatrix}0_{1\times n_\pi} & 1\end{bmatrix}\begin{bmatrix}W^\ell S^\ell & b^\ell\\ 0 & 1\end{bmatrix}\\ &:= F^\top F - ee^\top, \end{aligned} \tag{22}$$

where

$$\begin{aligned} F &= \begin{bmatrix}0_{n_{\pi\times(N_\mathbf{z}-N_\ell)}} & UW^\ell & Ub^\ell+V\end{bmatrix},\\ e &= \underbrace{\begin{bmatrix}0,\ 0,\ 0,\ ..,\ 1\end{bmatrix}^\top}_{N_\mathbf{z}+1}. \end{aligned} \tag{23}$$

Through the application of the Schur complement on M_{out}, the matrix inequality in (20) can be written as (18), which is linear in (V, U, Q). ■

Using the result of Theorem 1, the tightest bound on $\pi(z^0)$ is obtained from the following semidefinite program,

$$\begin{cases}\min_{U,V,Q} \ -\log\det(U) \text{ or } -\operatorname{tr}(U)\\ \quad \text{s.t.} \quad M(\tau, Q, U, V) \preceq 0,\\ \qquad \tau \geq 0,\ Q \in \mathcal{Q} \text{ from [9] (Lemma 4)},\end{cases} \tag{24}$$

where the objective function can be any metric related to the ellipsoid volume.

A. Application to Prediction Bound Propagation

We now apply the general multiple ellipsoid input robustness bound developed in Theorem 1 to find the prediction bound $\mathcal{E}(\hat{\mu}_{k+1}, \hat{\Sigma}_{k+1})$ with input ellipsoids $\mathcal{E}(y_{k-N}, \bar{\Sigma}_v)$, $\cdots$, $\mathcal{E}(y_k, \bar{\Sigma}_v)$.

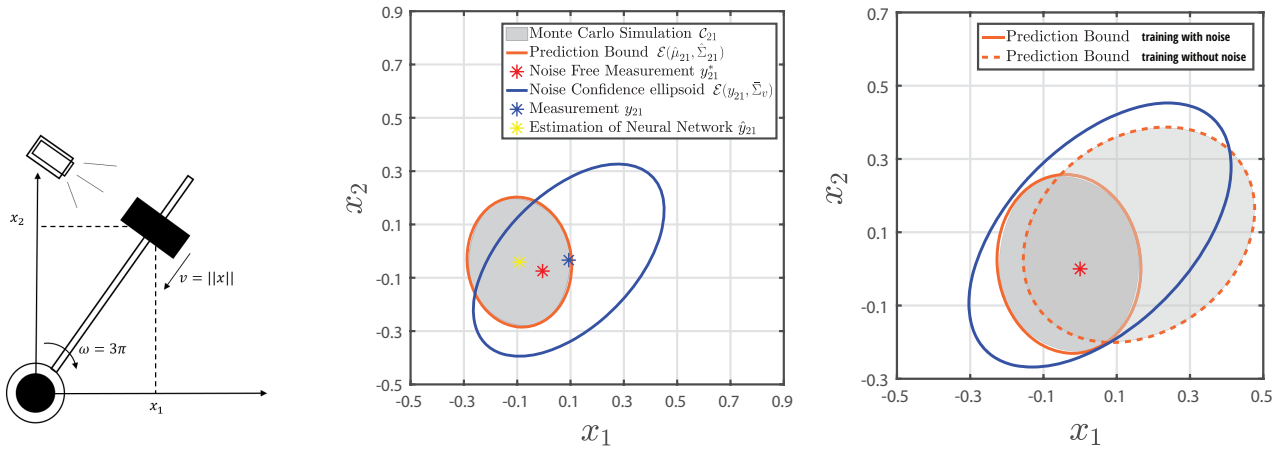
Proposition 2: Let $\gamma_{i,k} \in \mathcal{E}(y_{k-i+1}, \bar{\Sigma}_v)$ for $i = 1, 2, \cdots, q = N+1$ and define $\hat{\Sigma}_{k+1} = U^{-2}$, $\hat{\mu}_{k+1} = -U^{-1}V$, and $\mathcal{C}_k$ as the actual (non-ellipsoidal) prediction set of the neural network. Then $\pi(z^0) \in \mathcal{C}_k$ is bounded by $\mathcal{E}(\hat{\mu}_{k+1}, \hat{\Sigma}_{k+1})$ where the tightest bound comes from the convex optimization (24).

V. Numerical Experiments

We consider a linear beam and slider and a nonlinear water tank cascade to demonstrate our proposed detector.

A. Linear Beam and Slider

In the beam and slider (see Fig. 2a), a sensor measures the position of the slider which slides along the beam with a rate equal to its distance from the origin. The beam rotates with constant angular velocity of $3\pi\frac{rad}{sec}$. The sensor noise



(a) Beam and slider

(b) $k = 20$ Prediction bound

(c) Robustness due to training

Fig. 2: (a) The linear beam and slider system. (b) The effectiveness of the prediction bound at $k = 20$. (c) A neural network trained with noisy data has better robustness characteristics than one trained with noiseless data and is quantified well by our prediction bounds.

is Gaussian distributed, $v_k \sim \mathcal{N}(0, \Sigma_v)$, and we truncate it with a $\bar{p} = 95\%$ confidence ellipse $\mathcal{E}(y_k, \bar{\Sigma}_v)$ with

$$\Sigma_v = \begin{bmatrix} 0.0214 & 0.0112 \\ 0.0112 & 0.0217 \end{bmatrix} \text{ and } \bar{\Sigma}_v = \begin{bmatrix} 0.1282 & 0.0671 \\ 0.0671 & 0.1300 \end{bmatrix}.$$

We train a NARX feed forward neural network with two hidden layers (10 neurons in the first layer, 2 neurons in the second) with ReLU activation functions from noisy sensor data. The data was generated through simulation of the beam and slider system based on the linear discrete time model,

$$\begin{cases} x_{k+1} = 0.8 \begin{bmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{bmatrix} x_k, \quad \beta = \dfrac{3\pi}{5}, \\ y_k = x_k + v_k. \end{cases} \tag{25}$$

We use the current measurement and one past measurement (i.e., $N = 1$) to build the NARX model (we still call it NARX because the neural network returns a nonlinear approximation of this linear model).

Figure 2b demonstrates a snapshot of the efficacy of our methods under normal operation at $k = 20$. The prediction bound $\mathcal{E}(\hat{x}_{21}, \hat{\Sigma}_{21})$ (orange ellipse) can be quite tight on the actual prediction set $\mathcal{C}_{21}$ (gray region), which is the set of all possible neural network output estimates under all possible combinations from the input ellipsoids. As designed, the estimate $\hat{y}_{21}$ (is not necessarily the center $\hat{\mu}_{21}$) and ideal measurement y^*_{21} (assuming good accuracy of the estimation) are within the prediction bound. The actual measurement y_{21} is within the geometric sum of the prediction bound (orange) and confidence ellipsoid (blue); in this case it is also within the prediction bound, as the noise realization is relatively small. Under normal operation we received 0.7% false alarms compared with the upper bound on the false alarm rate $\bar{\mathcal{A}}_s = 1 - \bar{p}^3 = 14.26\%$. This conservatism below the upper bound illustrates that the distribution within the prediction boundary, and the subsequent geometric sum, is an important factor for improving our ability to predict the false alarm rate and an important direction for future work.

Despite the conservatism, we demonstrate that it is an effective tool for detecting anomalies in behavior. The first fault we consider is a vibration generated on the shaft because of the rotor. This imposes a new additive periodic displacement on the shaft and consequently the slider in a fixed direction

$$\delta x_k = \begin{bmatrix} \delta_1 \\ \delta_1 \end{bmatrix} \sin(k), \tag{26}$$

where $\delta_1 = 0.3$ and k is the time index. Under this fault the alarm rate raises to 27.17%.

The second scenario we consider is a sensor anomaly in which the sensor measurement is displaced by $\delta_2 = 0.3$,

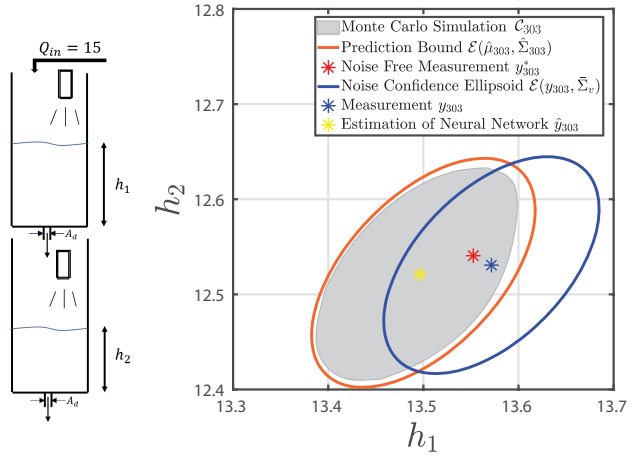
$$y_k = x_k + v_k + \begin{bmatrix} \delta_2 \\ \delta_2 \end{bmatrix}. \tag{27}$$

Under this fault the alarm rate raises to 25.35%.

The prediction bound provides a quantification of the robustness of a neural network to perturbed input. It is intuitive that a neural network trained by noisy data should have enhanced robustness compared with a neural network trained only by ideal (no noise) sensor measurements since the model parameters are learned to filter the noise more effectively. The prediction bound allows us to characterize this intuitive relationship quantitatively. In Fig. 2c, we provide the prediction bounds for this system at exactly identical conditions, with the only change that one model is trained with the actual noisy data and the other uses the underlying ideal (no noise) outputs. The larger prediction ellipsoid bound for the neural network trained without noise (dashed ellipse) quantifies the value of training with noisy data.

B. Nonlinear Water Tank Cascade

We now consider a two tank cascade (see Fig. 3a) in which flow out of the tanks is driven by gravity and yields



(a) Tank System (b) $k = 302$ Prediction Bound

Fig. 3: (a) The nonlinear water tank system. (b) The effectiveness of the prediction bound at $k = 302$.

a nonlinear dynamic,

$$\begin{cases} \dot{h_1}(t) = Q_{in} - c_d A_d \sqrt{2gh_1(t)}, \\ \dot{h_2}(t) = c_d A_d \sqrt{2gh_1(t)} - c_d A_d \sqrt{2gh_2(t)}, \\ y_k = h(t_k) + v_k. \end{cases} \tag{28}$$

where $t_k = k\Delta t$ with inter-sampling period $\Delta t = 0.02$ sec, $Q_{in} = 15$, $c_d = 0.9$, $A_d = 1$ cm^2, and g is the gravitational constant. For simplicity we use the same noise distribution and confidence level $\bar{p}$ as the prior example.

We train the neural network (two hidden layers: 20 neurons in the first layer; 5 neurons in the second layer) with the current noisy measurement and three (i.e., $N = 3$) past noisy measurements generated by simulating the above equations.

Figure 3b provides a snapshot of the performance of our approach under normal operation, at $k = 302$. Under normal operation the detector generates zero false alarms compared with the upper bound of $\bar{A}_s = 1 - \bar{p}^{N+2} = 22.62\%$.

Here we simulate a fault in the drainage of the lower tank (e.g., debris caught in the drain) such that 20% of the lower drain area is blocked. In this case, our proposed detector generates alarms at a rate of 58.1% in steady state.

VI. Conclusion

When first principle models of dynamical systems are too difficult to construct there is a growing tendency to turn to data-driven techniques. In this paper, we leverage recent results on the robustness of neural network predictions under input perturbations to compute bounds on the estimates produced from noisy measurements when a neural network is used to approximate an autoregressive model of the output measurements. We use this bound to define normal behavior under typical measurement noise and use this as a boundary to detect abnormal behavior.

This detector performs well and provides tight detection for the extreme points of normal behavior. This study has revealed that despite the tightness on these extreme points, a more accurate picture of the distribution of normal behavior is needed to produce tighter alarm rates. Nonetheless, we demonstrate the proposed detector is able to identify anomalies of a variety of types in a systematic way.

References

[1] M. H. Hassoun *et al.*, *Fundamentals of artificial neural networks*. MIT press, 1995.

[2] W. Liu, "An extended kalman filter and neural network cascade fault diagnosis strategy for the glutamic acid fermentation process," *Artificial intelligence in engineering*, vol. 13, no. 2, pp. 131–140, 1999.

[3] A. T. Vemuri and M. M. Polycarpou, "Neural-network-based robust fault diagnosis in robotic systems," *IEEE Transactions on neural networks*, vol. 8, no. 6, pp. 1410–1420, 1997.

[4] M. Wlas, Z. Krzeminski, J. Guzinski, H. Abu-Rub, and H. A. Toliyat, "Artificial-neural-network-based sensorless nonlinear control of induction motors," *IEEE Transactions on Energy Conversion*, vol. 20, no. 3, pp. 520–528, 2005.

[5] P. M. Frank and B. Köppen-Seliger, "New developments using ai in fault diagnosis," *Engineering Applications of Artificial Intelligence*, vol. 10, no. 1, pp. 3–14, 1997.

[6] Z. Li, L. Ma, and K. Khorasani, "Fault detection in reaction wheel of a satellite using observer-based dynamic neural networks," in *International Symposium on Neural Networks*. Springer, 2005, pp. 584–590.

[7] A. Abbaspour, P. Aboutalebi, K. K. Yen, and A. Sargolzaei, "Neural adaptive observer-based sensor and actuator fault detection in nonlinear systems: Application in uav," *ISA transactions*, vol. 67, pp. 317–329, 2017.

[8] M. Fazlyab, M. Morari, and G. J. Pappas, "Probabilistic verification and reachability analysis of neural networks via semidefinite programming," *arXiv preprint arXiv:1910.04249*, 2019.

[9] ——, "Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming," *arXiv preprint arXiv:1903.01287*, 2019.

[10] S. Dutta, S. Jha, S. Sanakaranarayanan, and A. Tiwari, "Output range analysis for deep neural networks," *arXiv preprint arXiv:1709.09130*, 2017.

[11] A. Lomuscio and L. Maganti, "An approach to reachability analysis for feed-forward relu neural networks," *arXiv preprint arXiv:1706.07351*, 2017.

[12] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International Conference on Machine Learning*, 2018, pp. 5286–5295.

[13] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, "Safety verification of deep neural networks," in *International Conference on Computer Aided Verification*. Springer, 2017, pp. 3–29.

[14] A. Raghunathan, J. Steinhardt, and P. S. Liang, "Semidefinite relaxations for certifying robustness to adversarial examples," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 877–10 887.

[15] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient smt solver for verifying deep neural networks," in *International Conference on Computer Aided Verification*. Springer, 2017, pp. 97–117.

[16] A. H. Abdelaziz, S. Watanabe, J. R. Hershey, E. Vincent, and D. Kolossa, "Uncertainty propagation through deep neural networks," 2015.

[17] J. S. Titensky, H. Jananthan, and J. Kepner, "Uncertainty propagation in deep neural networks using extended kalman filtering," *arXiv preprint arXiv:1809.06009*, 2018.

[18] C. Murguia and J. Ruths, "On model-based detectors for linear time-invariant stochastic systems under sensor attacks," *IET Control Theory & Applications*, vol. 13, no. 8, pp. 1051–1061, 2019.

[19] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[20] N. Hashemi and J. Ruths, "Co-design for performance and security: Geometric tools," *arXiv preprint arXiv:2006.08739*, 2020.

APPENDIX I
COMPARISON OF MULTIPLE INPUT APPROACH WITH SINGLE INPUT APPROACH

Consider the multiple ellipsoidal bounded inputs $\gamma_{i,k}$ introduced in Lemma 1. The quadratic constraint for this ellipsoid results to matrix M_i and the sum $\sum_{i=1}^{q} \tau_i M_i$ can be written as (suppressing the k index for readability)

$$\begin{bmatrix} -\tau_1\Sigma_{\gamma_1}^{-1} & 0 & \cdots & \tau_1\Sigma_{\gamma_i}^{-1}\mu_{\gamma_1} \\ 0 & -\tau_2\Sigma_{\gamma_2}^{-1} & \cdots & \tau_2\Sigma_{\gamma_2}^{-1}\mu_{\gamma_2} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_1\mu_{\gamma_1}^{\top}\Sigma_{\gamma_1}^{-1} & \tau_2\mu_{\gamma_2}^{\top}\Sigma_{\gamma_2}^{-1} & \dots & -\sum_{i=1}^{q}\tau_i\big(\mu_{\gamma_i}^{\top}\Sigma_{\gamma_i}^{-1}\mu_{\gamma_i}-1\big) \end{bmatrix}. \tag{29}$$

On the other hand if we follow [8] and define an overall ellipsoidal bound on the input vector, this bound should satisfy,

$$\sum_{i=1}^{q}(\gamma_{i,k}-\mu_{\gamma_{i,k}})^{\top}\Sigma_{\gamma_{i,k}}^{-1}(\gamma_{i,k}-\mu_{\gamma_{i,k}}) \leq q \tag{30}$$

(compare to (15)) where the equality happens where all the points $\gamma_{i,k}$ are selected from the boundary of the input ellipsoidal bounds. Constructing $\sum_{i=1}^{q} \tau_i M_i$ in this case has only one term so $q = 1$ and $\tau_1 = \tau$ leading to

$$\tau\begin{bmatrix} -\Sigma_{\gamma_1}^{-1} & 0 & \cdots & \Sigma_{\gamma_i}^{-1}\mu_{\gamma_1} \\ 0 & -\Sigma_{\gamma_2}^{-1} & \cdots & \Sigma_{\gamma_2}^{-1}\mu_{\gamma_2} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{\gamma_1}^{\top}\Sigma_{\gamma_1}^{-1} & \mu_{\gamma_2}^{\top}\Sigma_{\gamma_2}^{-1} & \dots & -\sum_{i=1}^{q}\mu_{\gamma_i}^{\top}\Sigma_{\gamma_i}^{-1}\mu_{\gamma_i}+q \end{bmatrix}. \tag{31}$$

Clearly (31) is a specific form of (29), in which the single decision variable τ in (31) is replaced with q decision variables $\{\tau_i\}_{i=1}^{q}$ in (29). Hence any solution for (31) can be expressed by (29) justifying that the multiple input ellipsoid approach can do no worse than the single input ellipsoid approach (i.e., it is as good or better than the single input approach).