Abstract—We propose a robust recurrent kernel online learning (RRKOL) algorithm based on the celebrated real-time recurrent learning approach that exploits the kernel trick in a recurrent online training manner. The novel RRKOL algorithm guarantees weight convergence with regularized risk management through the use of adaptive recurrent hyperparameters for superior generalization performance. Based on a new concept of the structure update error with a variable parameter length, we are the first one to propose the detailed structure update error, such that the weight convergence and robust stability proof can be integrated with a kernel sparsification scheme based on a solid theoretical ground. The RRKOL algorithm automatically weights the regularized term in the recurrent loss function, such that we not only minimize the estimation error but also improve the generalization performance through sparsification with simulation support.

Index Terms—Kernel online learning, real-time recurrent learning (RTRL), sparsification, weight convergence and robust stability.

I. INTRODUCTION

The basic idea of kernel trick has been used extensively in recent years [1]–[8]. The linear recursive least-squares algorithm [4], [5], [7] is a popular and practical algorithm that is used extensively in signal processing, communications, and control. The algorithm is an efficient online method for finding the least-squares linear predictor.

In online kernel learning applications, sparsification is performed at each time step using a new training sample, and a simple decision has to be made either add the sample into the representation or discard it. Engel et al. [3] and Richard et al. [7] propose treating the issues of regularization and computational efficiency using a type of online constructive sparsification algorithm based on the approximate linear dependence (ALD) and coherence criteria, while Liu et al. [4] proved that the kernel least mean square (KLMS) does not require regularization [4].

Real-time recurrent learning (RTRL)-based recurrent neural networks (RNNs) have been the topic of active research in recent years, and they have been proposed as an efficient technique for implementing nonlinear adaptive filtering because of their promising ability to model nonlinear dynamical systems [8]–[16]. Numerous applications can be found in various disciplines, especially for nonlinear time-series prediction and filtering problems [1]–[20].

In this paper, we present a new robust recurrent kernel online learning (RRKOL) algorithm, which is based on [13]. In addition to some minor changes of the original algorithm to match the theoretical proof as compared with [13], a new error analysis for the structural update of kernel online learning is presented, such that a complete weight convergence and the theoretical robust stability analysis based on sector theorem are derived for the RRKOL algorithm, which were absent in the conference version [13]. Furthermore, extended simulations up to three examples are presented in this paper to replace the original two examples. The idea behind RRKOL is to treat the kernel online training as an RTRL-based algorithm, in which the training depends not only on the current output of the network but also on the previous outputs, which is known as the Jordan network [8]. This dependence is particularly important for online training or real-time regression problems. There are two general approaches (with many variations) to the gradient and Jacobian calculations required for dynamic RNNs: backpropagation through time [9], [10] and RTRL [11], [12], [14]. The RTRL algorithm is also referred to as forward propagation or forward perturbation and is very efficient for online training [8], [14] and [21], which is used in this paper.

The integration of kernel and RTRL learning algorithms has historically attracted research interest, because both types of algorithms are powerful tools for the implementation of adaptive nonlinear time-series prediction and filtering algorithms [20], [24]. This integrated approach requires thorough research to simultaneously address two important issues in a unified manner, i.e., the sparsification of the kernel-based method and weight convergence and robust stability of the RTRL-based approach [5], [14], [24]. Previous research has addressed the two problems separately, for example, different kernel sparsification algorithms were developed, named the kernel recursive least squares (KRLS) algorithm [3], the quantized KLMS (QKLMS) algorithm [5], and the kernel-normalized least-mean-square (KNLMS) algorithm [7]. Novel adaptive nonlinear RTRL learning algorithms were proposed in [21]–[23] for adaptive online learning. A quantized KLMS algorithm subjected to a type of feedforward network weight convergence and robust stability analysis is presented in [5]. There is little research focused on a unified approach that considers the two important aspects of kernel sparsification and weight convergence with robust stability analysis for recurrent kernel online learning algorithms. The second issue is more important for the kernel recurrent learning algorithm compared with a normal feedforward network.

The concept of the structure update was first proposed in [5]. This concept should be linked to the variable length...
of the kernel vector function as introduced from the related kernel growing algorithms [3]–[5]. Note that the kernel online learning-based algorithm normally has only a single layer of neurons and, therefore, the structure update of kernel online learning is simply to decide the number of neurons in the single-output layer. It is interesting to note that this single-output layer has the output recurrent feedback loop in the proposed RRKOL algorithm, known as the Jordan network [8], which is simpler than and different from a normal multilayered RNN [14], [20]. However, the details of how the structure update can be analyzed through the weight convergence proof are not given in the previous research like [5]. To the best of our knowledge, we believe that we are the first one to propose a new concept of the structure update error, such that weight convergence and robust stability proofs, presented in this paper, are integrated with a kernel sparsification scheme with variable length. Another novel contribution of this paper is to establish an inherent link between a popular online kernel and RTRL learning methods through the implementation of the proposed RRKOL algorithm. We design an adaptive recurrent hyper-parameter with a switch ON–OFF condition for the recurrent training signal such that the method not only ensures weight convergence and robust stability but also yields kernel sparsification conditions that can automatically eliminate or stop the growth of the kernel function. Therefore, the RRKOL algorithm automatically weighs the regularized term in the loss function, such that we not only minimize the estimation error but also improve the generalization performance through sparsification with a rigorous theoretical proof. We extend the obtained $L_2$ stability result further to the $L_\infty$ robust stability condition as a result of the exponentially weighted delays up to $t$ fed back to the input side of the RNN with a discrete-time step $t$. The kernel expansion $K_t$ is defined in the vector form, with the variable kernel size $m_t$ at the discrete time step $t$, as follows:

$$K_t = [k(x_t, c_{1,t}), \ldots, k(x_t, c_{m_t,t})]^T \in R^{m_t \times 1} \quad (3)$$

where $k(x_t, c_{i,t})$ is the classical Gaussian kernel, $k(x_t, c_{i,t}) = \exp(-\|x_t - c_{i,t}\|^2/2\sigma^2)$, $i = 1, \ldots, m_t$, $\sigma > 0$ is the kernel width, and $c_{i,t}$ with $1 \leq i \leq m_t$ is the center of the Gaussian kernel function.

The single output $y_t$ of the RNN with a kernel function in the proposed algorithm can typically be presented as a reduced radial basis function (RBF) or RBF-like network of size $m_t$ at the discrete time step $t$ of the recurrent output feedback of a Jordan network [8]

$$y_t = \alpha_t K_t = \sum_{i=1}^{m_t} \alpha_{t,i} k(x_t, c_{i,t}). \quad (4)$$

Note that in this paper, we are interested in online learning, as shown in Fig. 1, where the examples become available one by one, and it is desired that the learning algorithm depends on a sequence of hypotheses $y_{t-1}, \ldots, y_{t-d}$. This type of learning framework is appropriate for real-time learning problems and is, of course, analogous to the usual adaptive signal processing framework [20]. Furthermore, the system input variable $u_t$ can also be used with delayed input steps up to $l$, as shown in Fig. 1 for the online training.

Because the number of adjustable linear parameters in kernel solutions equals the kernel expansion size (3), one must introduce some form of regularization [2]. In addition to the recently developed information-theoretic approach, which can be viewed as an extension of ALD [4], there are two primary existing criteria in the online kernel learning literature: 1) the novelty criterion proposed by Platt in resource-allocating networks and 2) the ALD test introduced in [3] for KRLS [3]. The prediction variance criterion is similar to the ALD-related sparsification condition, in which a new kernel function $k(\cdot, x_t)$ at step $t$ is inserted into the dictionary if the following condition is satisfied. Note that the dictionary is referred to the list of centers $c_{i,t}$ ($1 \leq i \leq m_t$) of all kernel function $k(x_t, c_{i,t})$ as defined in (4) of this paper and mentioned in [3] and [4]. The ALD sparsification is presented as

$$1 - K_t^T G_t^{-1} K_t \geq \zeta \quad (5)$$
where $\zeta$ is a positive number that determine the level of sparsity of the model. $G_t$ is the kernel Gram matrix

$$G_t = \begin{bmatrix} k(c_{t,1}, c_{t,1}) & \cdots & k(c_{t,1}, c_{t,m_t}) \\ k(c_{t,2}, c_{t,1}) & \ddots & \vdots \\ \vdots & \ddots & k(c_{t,m_t-1}, c_{t,m_t}) \\ k(c_{t,m_t}, c_{t,1}) & \cdots & k(c_{t,m_t}, c_{t,m_t}) \end{bmatrix}. \quad (6)$$

Because the evaluation of the ALD sparsification condition, especially for the inverse of $G_t$, is very expensive in (5), Richard et al. [7] proposed a simplified coherence criterion that is an approximation to the ALD-related sparsification condition (5); the coherence sparsification criterion is given in the following equation:

$$\max |k(c_{t,i}, x_i)| \leq \mu_0$$

where $1 \leq i \leq m_t$, $\mu_0 \in (0, 1)$ is a positive parameter that determines both the level of sparsity and the coherence of the dictionary.

Based on ALD or coherence-related sparsification conditions, the selected feature sequence $\{\phi(c_{t,i})\}$ $1 \leq i \leq m_t$ forms a dictionary with the linearly independent centers in the feature space for popular kernel online learning algorithms [4], [7]. To the best of our knowledge, there is no previous research that presents a weight convergence and robust stability study that is integrated with either ALD or a coherence sparsification-related algorithm [3], [7], particularly for recurrent kernel learning.

### B. RRKOL Algorithm and Modeling

We present a new family of recurrent kernel online learning algorithms to naturally translate kernel online learning into recurrent learning in an RTRL manner. We first discuss the case in which the structural risk function is determined using recurrent constraint learning, and focus on a cost function that determines the empirical risk function regulated by a recurrent sparsification scheme. The risk function that we define is a regularized least-mean-squares function in the sense that it not only minimizes the training error in the conventional cost function but is also integrated with the sparsification process. Note that we base our model on the popular kernel least-squares growing algorithm [1], [4], [5], in which the number of kernel functions increases by one at each iteration of the proposed RRKOL algorithm if the sparsification and weight convergence and robust stability conditions (specified below) are met simultaneously. This requires that we limit the complexity of the network or the number of kernel functions to reduce the computational cost of the algorithm and, more importantly, to improve the generalization performance to avoid the overfitting problem.

1) **Optimal or Suboptimal Models With Different Loss Functions:** Related to model (4) with the unknown parameter weight vector and kernel function centers, the optimal or suboptimal or ideal system output $y^*_t$ in the proposed algorithm can typically be presented as

$$y^*_t = a^*, K^*_t = \sum_{i=1}^{m^*} a^*_i k(x_t, c^*_i) \quad (8)$$

where the optimal or suboptimal or ideal coefficient weight vector $a^*$ with an optimal or suboptimal fixed size $m^*$ is expressed as

$$a^* = [a_1^*, \ldots, a_{m^*}^*] \in \mathbb{R}^{1 \times m^*}. \quad (9)$$

Note that the optimal or suboptimal terms refer to the best possible model (8), which can represent the real-world subject, for example, the real-time series in the prediction problem with these optimal or suboptimal parameters. It not only minimizes the instant loss function as proposed later in the section, but also provides the best possible generalization performance for time-vary and unmodeled dynamics, known as generalization performance [20], [23].

Similarly, the optimal or suboptimal kernel function center is presented as $c_{i^*}^*$ with $1 \leq i \leq m^*$ with the optimal or suboptimal kernel expansion

$$K^*_t = [k(x_t, c_{1}^*), \ldots, k(x_t, c_{m^*}^*)]^T \in \mathbb{R}^{m^* \times 1}. \quad (10)$$

Note that the optimal or suboptimal kernel center $c_{i^*}^*$ may be estimated by a nonlinear gradient iteration algorithm similar to the optimal or suboptimal weight $a_{i^*}^*$. However, in this paper, we use the ALD-based growing algorithm to select the best available observation data according to ALD-related sparsification criterion to approximate the optimal or suboptimal centers with a variable kernel number $m_t$ of the kernel function $K_t$ as defined in (3) of this paper with reference [7]. Therefore, the optimal or suboptimal center $c_{i^*}^*$ is approximated by $c_{i^*}$ in the recurrent output of the estimation model (4) through the coherence sparsification condition (7). Furthermore, we use the conic sector theorem to relax convergence and robust stability conditions of the RRKOL in the $L_\infty$ space (see Section III for more details) [25], [26]. The optimal or suboptimal model means that the ideal center $c_{i^*}^*$ is selected in the sense of ALD-related criterion (7). $c_{i^*}^*$ could also be selected via other criterion, such that the ideal model output $y_{t^*}^*$ in (8) may not be unique optimal.

We can define the conventional instantaneous estimation error

$$e_t = y_{t^*}^* - y_t + e_{i^*}^0 \quad (11)$$

where $e_{i^*}^0$ represents the system disturbance and the neural network approximation error [20].

In a normal least-mean-squares algorithm, the parameter weight vector $a_t$ can easily be estimated through a conventional feedforward gradient (derivative) iterative equation

$$a_{t+1} = a_t - \eta F \frac{\partial E_t}{\partial a_t}$$

$$= a_t + \eta F e_t K_t^T \quad (12)$$

where $\eta F$ is a constant learning rate and the instantaneous loss function is

$$E_t^I = \frac{1}{2}(e_t)^2. \quad (13)$$

However, there are two problems with the iterative equation (12). First, the classical estimate algorithm is sensitive to disturbance $e_{i^*}^0$, based on a relatively large constant learning rate $\eta F$, and the learning process can be very slow if the
learning rate $\eta^F$ is too small. The second problem is setting up an iterative equation based on a nonlinear gradient of $E^f_t$, as in (12), to estimate the kernel function center $c_{i,t-1, \ldots, c_{i,t}}$. It may be estimated in a manner similar to that in (12). It will lead to a nonlinear recursive estimation problem without capability to control the size of the network [20], [21]. Notice that in a normal feedforward neural network, the instant estimation error $e_t$ only depends on the current network output $y_t$ and the teaching (supervision) signal $y^*_t$; therefore, the gradient term for the online training will simply be $(\delta e^2_t / \delta a_t)$, which is not suitable for kernel online training and prediction purposes. A more suitable measure of performance for online algorithms in an online setting is the cumulative loss function

$$E^C_t = \frac{1}{2} \sum_{i=1}^{d} (e_i)^2$$

where $d$ is the cumulative step. It is interesting to note that the recurrent structure of the RRKOL algorithm in Fig. 1 is also based on the $d$ steps delayed output signal to predict the future output with the updated weight vector $a_{t+1}$. The recurrent feedback signals $y_{t-1}, \ldots, y_{t-d}$ form the best base using past patterns to predict the future output. Notice that some researchers also use the delayed input patterns $u_{t-1}, \ldots, u_{t-d}$ for training; however, this approach is not as effective as using the delayed feedback output signal, as in the recurrent structure shown in Fig. 1 [20].

If we can guarantee a low cumulative loss, we are already guarding against overfitting, i.e., limiting the number of kernel functions to a small size [1], [20]. However, regularization can still be useful in the online setting: if the target we are learning changes over time, regularization prevents the hypothesis from going too far in one direction, thus hopefully helping recovery when a change occurs. Furthermore, if we are interested in large-margin algorithms, some types of complexity control are needed to make the definition of the margin meaningful.

The RTRL training is based on the concept of utilizing the current training error together with the extra information provided by the past recurrent feedback signal $y_{t-1}, \ldots, y_{t-d}$ [16], [20]. Therefore, for the reduced data dictionary, we introduce the RRKOL algorithm in this paper as a recurrent version of the kernel online learning. The RRKOL training depends on the recurrent output $y_{t-1}, \ldots, y_{t-d}$ of the RNN in addition to the instantaneous training error $e_t$ to predict the future (the next discrete-time step) kernel weight vector $a_{t+1}$. We still use the instantaneous estimation error $e_t$ to simplify the online learning process, but instead of minimizing the cumulative loss function like (14) directly, we introduce the regularized risk term in the differential function of the recurrent loss function related to not only the instantaneous error $e_t$ (and the current RNN output $y_t$) but also the recurrent feedback signal $y_{t-1}, \ldots, y_{t-d}$ up to the last $d$ steps.

In other words, we use the full derivative of the instantaneous loss function $E^f_t$ according to the chain rule [in contrast to the partial derivative function used in the feedforward training iteration (12)]

$$\frac{dE^f_t}{da_t} = \frac{1}{2} \left[ \frac{\partial (e^2_t)}{\partial a_t} + \frac{\partial (e^2_t)}{\partial K_t} \frac{\partial K_t}{\partial a_t} \right].$$

Because the second recurrent term in (15) causes extra weight convergence and robust stability issues (see Section III for more details), we introduce an important adaptive recurrent hyperparameter matrix $\lambda_t$ (defined below) to weight the second recurrent term, such that (15) becomes

$$\frac{dE^f_t}{da_t} = \frac{1}{2} \left[ \frac{\partial (e^2_t)}{\partial a_t} + \frac{\partial (e^2_t)}{\partial K_t} \frac{\partial K_t}{\partial a_t} \right]$$

$$= -e_t (K_t^T + a_t K''_t D^a_t \lambda_t)$$

(16)

where

$$K''_t = \begin{bmatrix} k(x_t, c_{i,1}) \left( -\frac{x_t - c_{i,1}}{\sigma^2} \right) \ldots k(x_t, c_{i,m}) \left( -\frac{x_t - c_{i,m}}{\sigma^2} \right) \end{bmatrix}^T$$

$$\in \mathbb{R}^{m_t \times n}$$

(17)

and $D^a_t$ is the RTRL-style iterative approximation term [11], [14]

$$D^a_t = \begin{bmatrix} \frac{\partial u_t}{\partial a_t}, \ldots, \frac{\partial u_t}{\partial a_t}, \frac{\partial y_{t-1}}{\partial a_t}, \ldots, \frac{\partial y_{t-d}}{\partial a_t} \end{bmatrix}$$

$$\approx \begin{bmatrix} \frac{\partial y_{t-1}}{\partial a_{t-1}} | 0 \ldots 0 | \ldots | \frac{\partial y_{t-d}}{\partial a_{t-d}} | 0 \ldots 0 \end{bmatrix} \in \mathbb{R}^{n \times m_t}.$$  

(18)

The approximation in (18) is based on the traditional RTRL approach in [14] and [20]. Because the derivative $(\partial y_{t-1} / \partial a_t)$, for example, cannot be used in the real-time calculation, therefore, an approximation

$$\left[ \frac{\partial y_{t-1}}{\partial a_t} \right]^T = \left[ \frac{\partial y_{t-1}}{\partial a_{t-1}} \right]^T \begin{bmatrix} 0 \ldots 0 \end{bmatrix} \in \mathbb{R}^{n \times m_t}.$$  

is used with extra zeros on the right side of the row $l + 1$ of the last approximation of (18) (note that the zeros are to make every row with the equal length). Similar approximation are used for the next $m_t - 1 - m_{t-d}$ rows. Please also note that this approximation similar to all RTRL-based algorithms as in [14] may cause extra estimation error of the RRKOL algorithm, which is one of the basic reasons that for the conic sector stability analysis later in Section III to study $L_\infty$ results with a nonzero equivalent disturbance.

The hyperparameter matrix $\lambda_t$ is defined according to the weight convergence and robust stability and sparsification conditions, as we show in Section III with

$$\lambda_t = \lambda^0_t K_t K_t^T \in \mathbb{R}^{m_t \times m_t}$$

(19)

$$\begin{cases} \lambda^0_t = \frac{1}{\mu_0 + 1}, & \text{if } (m_t - 1)\mu_0 < 1 \\ \lambda^0_t = 0, & \text{if } (m_t - 1)\mu_0 \geq 1. \end{cases}$$

(20)
The switch ON–OFF condition \((m_t - 1)\mu_0\) is selected, such that an optimal or suboptimal compromise between the weight convergence and robust stability and an approximation of linearly dependent sparsification, such as the coherence condition (7), is achieved (see remark 2 and the proof of Proposition 3 in Section III). \(\lambda_0\) is a positive constant, and 
\[
\Gamma_t = \left((|K_t||G_t||A_t|_{1,1}1/[(m_t - 1)\mu_0])\right),
\]
where we use \(||.||\) to denote the \(L1\) vector norm and \(||.|_{1,1}\) to denote the \(L2\) matrix norm. \(||.|\) denotes the \(L2\) norm by default in the rest of this paper. Note that the optimal or suboptimal value of the recurrent constrained hyperparameter matrix \(\lambda_t\) must be selected appropriately according to the weight convergence and robust stability and sparsification conditions, as we show in Section III.

Remark 1: The two equalities of (16) are derived through the kernel network function (vector) \(K_t\) in (3) according to the chain rule of the derivative function. These lead to \((1/2)(\partial(e^2_t)/\partial K_t) = e_t\alpha_t\) and \((\partial K_t/\partial e_t) = K_t''D_t^{\alpha_t}\), as in (16). The dependence of the recurrent feedback signals \(y_{t-1,\ldots,y_{t-d}}\) with respect to the current estimate vector \(\alpha_t\) is called the nonlinear parallel leaning configuration in the classical RNN approaches [16], [20]. However, the derivative-related dependence term \(D_t^{\alpha_t}\) in the first row of (18) is difficult for online implementation; therefore, an approximation in the second row of (18) is used for the popular RTRL learning, as adopted by many previous researchers in the area of RNNs [16], [20], [22], [23].

2) RRKOL Algorithm: To accommodate the changeable dimension of the kernel expansion \(K_t\) in (4) of the RRKOL algorithm, which is increased by one \(m_{t+1} = m_t + 1\), which depends on the coherence-related sparsification condition (7), as presented in the algorithm at the end of this section, i.e.,

\[
\bar{\alpha}_t = [\alpha_t,0], \quad \bar{K}_t = [K_t^{T} 1], \quad \bar{D}_t^{\alpha} = [D_t^{\alpha} \text{Col}[0]], \quad \bar{K}_t'' = [K_t'' \text{Col}[0]]^T, \quad \bar{\zeta}_t = \lambda_t^2 \bar{K}_t \bar{K}_t^T, \quad \bar{A}_t = \bar{\alpha}_t \bar{K}_t \bar{D}_t^{\alpha},
\]

with
\[
A_t = \alpha_t K_t'' D_t^{\alpha} \in R^{1 \times m_t}. \tag{21}
\]

Otherwise, we have the same dimension \(m_{t+1} = m_t\) with \(\bar{\alpha}_t = \alpha_t, \quad \bar{K}_t = K_t, \quad \bar{D}_t^{\alpha} = D_t^{\alpha}, \quad \bar{K}_t'' = K_t'', \quad \bar{\zeta}_t = \lambda_t, \quad \text{and} \ \bar{A}_t = A_t.\)

The RRKOL weight iterative equation is

\[
\alpha_{t+1} = \bar{\alpha}_t + \frac{\eta_t}{\rho_t} e_t(\bar{K}_t^{T} + \bar{\alpha}_t \bar{K}_t \bar{D}_t^{\alpha} \bar{\lambda}_t) \tag{22}
\]

\[
\bar{\alpha}_{t+1} = \bar{\alpha}_t + \frac{\eta_t}{\rho_t} e_t(\bar{K}_t^{T} + \bar{A}_t \bar{\lambda}_t).
\]

The adaptive normalization factor is defined as in [25] and [26]
\[
\rho_t = \nu\rho_{t-1} + \max \left\{ \frac{\rho_t}{\sqrt{2} \bar{\lambda}_t^{T} \bar{A}_t \bar{K}_t \bar{K}_t^{T}}, 1 + \lambda_t^2 \bar{K}_t^{T} \bar{\lambda}_t \right\}. \tag{23}
\]

with \(\nu > 0\) and \(\bar{\rho} > 0\).

The classical adaptive learning rate is defined to counteract the weight drift in the iteration [23]
\[
\eta_t = \begin{cases} 
\eta & \text{if } |e_t| \geq \varepsilon_m / \sqrt{T} \\
0 & \text{if } |e_t| < \varepsilon_m / \sqrt{T}
\end{cases} \tag{24}
\]

Algorithm

- At time \(t = 1\), take the first pattern \(x_1\) and initialize \(c_{1,1}, K_1, D_1^{\alpha}, K_1''\), with \(m_1 = 1\).
- At time \(t = t + 1\), temporarily set \(m_t = m_t + 1\) and input a new online datum \(x_t\). If \(\max(|c_{t,1}, x_t|) \leq \mu_0\) with \(1 \leq t \leq m_t\), then take \(x_t\) into the dictionary as a future kernel center for the next growing step and perform the following assignments:
  
  - \(\bar{D}_t^{\alpha} = [D_t^{\alpha} \text{Col}[0]]\)
  - \(\bar{\alpha}_t = [\alpha_t, 0]\)
  - else, set \(m_t = m_t\) and ignore the current input datum as in the dictionary.
- Update \(K_1, K_t', D_t^{\alpha}, A_t\) using the new datum \(x_t\).
- Update
  \[
  a_{t+1} = \bar{\alpha}_t + \eta_t \bar{e}_t(\bar{K}_t^{T} + \bar{A}_t \bar{\lambda}_t) \tag{25}
  \]
  and then return to the second step.

in which the constant satisfies \(\varepsilon_m > 0, \eta > 0, \) and \(\sqrt{T} > 0\) (see Section III for details).

The RRKOL algorithm is similar to the popular kernel-based online training algorithms that increase the number of kernel functions by one in each step if the new data are not associated with the old data in terms of the ALD condition [1], [4]. We use the coherence-based kernel sparsification method in [7], which is computationally more efficient compared with the standard ALD-based algorithm [4]. The RRKOL algorithm guarantees weight convergence and robust stability and, therefore, an overall smoother time response and a lower computational time requirement compared with other kernel-based algorithms. The RRKOL algorithm with a coherence-based sparsification scheme is defined in Algorithm 1.

III. SPARSIFICATION, WEIGHT CONVERGENCE AND ROBUST STABILITY: AN INTEGRATED THEORETICAL ANALYSIS

We show that the proposed RRKOL algorithm can be naturally derived from the weight convergence and robust stability condition and is connected to the coherence-based sparsification condition. Furthermore, we present some closed formulas for the sparsification scheme that can be derived from the RBF-like recurrent network as integral expressions. By focusing on the Gram matrix embedded in the weight convergence and robust stability proof, we provide explicit formulas suited for the design of the robust recurrent training algorithm.

A. Error Analysis and Weight Convergence

To start the theoretical proof and analysis, we define the weight error sector of the kernel function as
\[
\bar{\alpha}_t = \bar{\alpha}_t - a^{m_{t+1}} = R^{1 \times m_{t+1}} \tag{26}
\]
with
\[
a^{m_{t+1}} = [\alpha_1^*, \ldots, \alpha_{m_{t+1}}^*] \in R^{1 \times m_{t+1}}. \tag{27}
\]
Note that we have normally $m_t \leq m^* \forall t$, since the relevant kernel online learning algorithms similar to this paper are normally starting from $m_t = 1$ at the initial time and then gradually approaching the optimal or suboptimal kernel number $m^*$ according to the coherence sparsification criterion (7) as in [3] and [4].

The error equation (11) can be rewritten in the form of a structure update error using the weight error vector term $\tilde{a}_t$ as

$$e_t = y^*_t - y_t + \tilde{e}_t^0$$

$$= y^*_t - \alpha_t K_t + \tilde{e}_t^0 = a^*_t K^* - \overrightarrow{a}_t \overrightarrow{K}_t + \tilde{e}_t^0$$

$$= a^{m_{t+1}} \overrightarrow{K}_{t}^{m_{t+1}} - \overrightarrow{a}_t \overrightarrow{K}_t + \sum_{j=m_{t+2}}^{m^*} a^*_j k(x_j, c^*_j) + \tilde{e}_t^0$$

$$= a^{m_{t+1}} \overrightarrow{K}_{t}^{m_{t+1}} - a^{m_{t+1}} K_t$$

$$+ \sum_{j=m_{t+2}}^{m^*} a^*_j k(x_j, c^*_j) + \tilde{e}_t^0$$

$$= \tilde{e}_t - \tilde{e}_t^0$$

with $K_t^{m_{t+1}} = [k(x_t, c^*_1), \ldots, k(x_t, c^*_m)]^T$ and

$$\tilde{e}_t^0 = \overrightarrow{a}_t \overrightarrow{K}_t - a^{m_{t+1}} \overrightarrow{K}_t = \tilde{a}_t \overrightarrow{K}_t$$

(28)

and

$$e_t^a = a^{m_{t+1}} \overrightarrow{K}_t - a^{m_{t+1}} \overrightarrow{K}_t = \tilde{a}_t \overrightarrow{K}_t$$

(30)

We name $e_t$ as an equivalent disturbance for the robustness analysis of the RRRKOL algorithm, because it includes both the system external disturbance and the kernel center modeling error $K_t^{m_{t+1}} - \overrightarrow{K}_t$ and $\sum_{j=m_{t+2}}^{m^*} a^*_j k(x_j, c^*_j)$, which may cause the weight convergence and stability issues of the RNN.

It is interesting to note that the concept of the structure update was first proposed in [5] which is linked to the variable length of the kernel vector $K_t$ as defined in (3). However, the details of how the structure update can be analyzed through the weight convergence proof are not given in the previous research like [5]. To the best of our knowledge, it is believed that we are the first one to propose the detailed structure update error as defined in (28), such that the following weight convergence proof can be integrated with the ALD-related sparsification scheme (7). In other words, a noise level of the equivalent disturbance $e_t$ can be reduced, because the first two terms of $e_t$ defined in (30) are gradually eliminated in the processing of the ALD-related sparsification scheme (7) as the kernel length $m_t$ is gradually increased from one to the optimal or suboptimal kernel number $m^*$, as presented in the RRRKOL algorithm in Section II.

Assume that we use the classical feedforward weight-updating equation (12), subtract the $a^{m_{t+1}}$, and square the both sides of (12), we have

$$\|\tilde{a}_{t+1}\|^2 - \|\tilde{a}_t\|^2 = 2\eta F e_t \tilde{a}_t \overrightarrow{K}_t + (\eta F)^2 \tilde{e}_t^0 \|\overrightarrow{K}_t\|^2.$$  

(31)

By multiplying both sides of (28) by $e_t$, we have $e_t e_t^a = e_t \tilde{e}_t - \tilde{e}_t^0$, and replacing $e_t e_t^a = e_t \tilde{a}_t \overrightarrow{K}_t$ with $e_t e_t - e_t^2$, we can rewrite (31) as

$$\|\tilde{a}_{t+1}\|^2 - \|\tilde{a}_t\|^2 = [2\eta F e_t \tilde{a}_t \overrightarrow{K}_t + (\eta F)^2 \tilde{e}_t^0 \|\overrightarrow{K}_t\|^2].$$

(32)

To ensure weight convergence and robust stability, the last equation of (32) must be less than or equal to zero, that is

$$\tilde{e}_t^2 \leq \frac{2\eta F e_t - e_t^2}{\eta F \|\overrightarrow{K}_t\|^2} \forall e_t.$$  

(33)

However, the convergence and robust stability condition of (33) cannot be guaranteed in presence of the noise term $e_t$, which includes both system disturbance $\tilde{e}_t^0$ and kernel center error-related term, as presented in (30). Therefore, for a fixed learning rate $\eta F \ll 1$, the kernel function network output $y_t$ may converge to a mapping function, which will be very different from the optimal or suboptimal kernel function output $y^*_t$ as in (8) (see the simulation in Section IV for details).

Motivated by the above analysis, we propose the adaptive learning system in the RRKOL algorithm with the following weight convergence and robust stability analysis.

As introduced in Section II, the inverse of the kernel Gram matrix (6) plays an important role in the ALD and coherence-related sparsification algorithms. Therefore, the inverse kernel matrix $[G_t]^{-1}$ can be embedded into an equation $1 + \lambda^2 K_t A_t > 0$, which is further linked to the coherence-based sparsification condition (7) and the weight convergence condition in Proposition 2 in the latter part of Section III. Note that this implies that we embed the kernel Gram matrix $G_t$ and its inverse implicitly into the recurrent hyperparameter $\lambda_t = \lambda^2 K_t A_t^T$, such that the regularized loss function (16) can address the sparsification requirement.

**Proposition 1:** The inequality $1 + \lambda^2 K_t A_t > 0$ in Proposition 1 is always true if the recurrent hyperparameter $\lambda^2$ can be defined as (20) and linked to the coherence parameter $\mu_0$.

**Proof:** According to the definition of $\lambda^2$ in (20), $1 + \lambda^2 K_t A_t > 0$ is true if $(m_t - 1)\mu_0 \geq 1$, i.e., $\lambda^2 = 0$, and thus, we need only prove the case of $(m_t - 1)\mu_0 < 1$ as follows.

As stated in Section II, the Gram kernel matrix $G_t$ uses the Gaussian kernel with a unit diagonal. We first rewrite the Gram kernel matrix $G_t$ using an off-diagonal component $\overrightarrow{G}_t$ that satisfies

$$G_t = I_m + \overrightarrow{G}_t.$$  

(34)

By the definition of the $L_1$ norm [7], [29] and letting $\mu_0 \in (0, 1)$ be a positive number as defined in (7), the $L_1$ norm of the matrix $\overrightarrow{G}_t$ is presented as

$$\mu_1 (m_t - 1) = || \overrightarrow{G}_t ||_{1,1} = \max_j \sum_{j \neq i} |k(c_{j,i}, c_{i,j})| \leq (m_t - 1)\mu_0.$$  

(35)
Note that the $L_1$ norm of the matrix $\overline{G}_t$ measures the maximum total coherence between a fixed center and a collection of others. However, due to the calculation complexity of $\mu_1(m_t - 1) < 1$, we may also use a more restrictive condition $(m_t - 1)\mu_0 < 1$ to replace it to reduce the computational cost [7]. Furthermore, it was argued in [7] that the $m_t$ kernel functions $k(\cdot, c_{t1}), \ldots, k(\cdot, c_{tm_t})$ from a dictionary become linearly independent if $\mu_1(m_t - 1) < 1$. However, we only need a more restrictive condition $(m_t - 1)\mu_0 < 1$ to define the recurrent hyperparameter $\lambda_t$ to ensure the weight convergence and robust stability condition, because the coherence sparsification condition, such as the ALD sparsification condition in (5), is just an approximation of a linearly independent condition that may be needed to eliminate more input data than the true linearly independent condition for an effective sparsification (see remark 4 later in this section for additional discussion).

Because we have $\lambda_t \neq 0$, i.e., $(m_t - 1)\mu_0 < 1 \Rightarrow ||G_t||_{1,1} < 1$, as shown in (35), we can further extend the inverse of the kernel Gram matrix as a von Neumann series and use Banach algebra methods to estimate its norm. Note that the von Neumann series $\sum_{k=0}^{\infty}(-G_t)^k$ under the inequality condition (35) converges to the inverse $[I_{m_t} + G_t]^{-1}$ [27], [29]. Thus

$$||G_t^{-1}||_{1,1} = ||[I_{m_t} + G_t]^{-1}||_{1,1}$$

$$= \sum_{k=0}^{\infty}||(-G_t)^k||_{1,1}$$

$$\leq \sum_{k=0}^{\infty}||G_t||_{1,1}^k = \frac{1}{1 - ||G_t||_{1,1}}$$

$$\leq \frac{1}{1 - (m_t - 1)\mu_0}. \quad \text{(36)}$$

According to (36), we can further develop

$$|\lambda_t^0 K_t^T A_t^*|$$

$$= |\lambda_t^0 K_t^T G_t[G_t]^{-1} A_t^*|$$

$$\leq |\lambda_t^0 K_t^T G_t G_t G_t^{-1} A_t^*| \leq \frac{\lambda_0^0 ||K_t||_1 ||G_t||_{1,1} ||A_t||_{1,1}}{1 - (m_t - 1)\mu_0} < 1$$

$$\Rightarrow 1 + \lambda_t^0 K_t^T A_t^* > 0. \quad \text{(37)}$$

**Proposition 2:** Assume that $||e_t|| < \varepsilon_m$ is a bounded equivalent disturbance in the $L_2$ space, as introduced in (30) (note that $e_t$ is indeed bounded if the external disturbance $e_t^0$ is bounded as usually in adaptive filtering applications). The weight of the kernel iterative equation (22) of the RRKOL algorithm converges to a constant value.

**Proof:** Subtract the $a^{sm_t+1}$ and square both sides of (22), we obtain

$$||\tilde{a}_{t+1}||^2 - ||\tilde{a}_t||^2$$

$$= 2\eta_t \rho_t e_t \tilde{a}_t (K_t + \lambda_t^0 K_t^T A_t^*) + \eta_t^2 e_t^2 ||K_t^T + \lambda_t^0 A_t^*||^2 / \rho_t^2. \quad \text{(38)}$$

The key difference between the recurrent and feedforward neural network training algorithms is the recurrent training term $A_t^T$ in (22), which is derived from the second term of the chain rule for the derivative (16) and is absent from the feedforward training in (12). To derive the Lyapunov condition from (38), i.e., to ensure that it is no larger than zero, we rewrite (38) as

$$||\tilde{a}_{t+1}||^2 - ||\tilde{a}_t||^2 = \left[2\eta_t \rho_t e_t \tilde{a}_t (K_t + \lambda_t^0 K_t^T A_t^*) + \eta_t^2 e_t^2 ||K_t^T + \lambda_t^0 A_t^*||^2 / \rho_t^2 \right]$$

$$= \left[2\eta_t \rho_t e_t \tilde{a}_t (1 + \lambda_t^0 K_t^T A_t^*) + \eta_t^2 e_t^2 ||K_t^T + \lambda_t^0 A_t^*||^2 / \rho_t^2 \right]. \quad \text{(39)}$$

Note that the last equality of the above equation is derived from (29).

By multiplying both sides of (28) by $e_t$, we have $e_t e_t^* = e_t e_t - e_t^2$, and replacing $e_t e_t^*$ with $e_t e_t - e_t^2$, we can rewrite (39) as

$$||\tilde{a}_{t+1}||^2 - ||\tilde{a}_t||^2 = \left[2\eta_t \rho_t (e_t e_t - e_t^2) (1 + \lambda_t^0 K_t^T A_t^*)ight.$$  

$$+ \eta_t^2 e_t^2 ||K_t^T + \lambda_t^0 A_t^*||^2 / \rho_t^2$$

$$\leq \left[\eta_t \rho_t (e_t^2 + e_t^2) (1 + \lambda_t^0 K_t^T A_t^*)ight.$$  

$$- 2\eta_t \rho_t e_t^2 (1 + \lambda_t^0 K_t^T A_t^*)$$  

$$+ \eta_t^2 e_t^2 ||K_t^T + \lambda_t^0 A_t^*||^2 / \rho_t^2 \right]. \quad \text{(40)}$$

Now, we can use the fact $(1 - \pi_t) > 0$ with $\pi_t = (\eta_t \rho_t ||K_t^T + \lambda_t^0 A_t^*||^2 / \rho_t (1 + \lambda_t^0 K_t^T A_t^*))$ according to the definition of the normalization factor $\rho_t$ in (23) to rewrite (40) as

$$||\tilde{a}_{t+1}||^2 - ||\tilde{a}_t||^2$$

$$\leq -\frac{\eta_t}{\rho_t} \left(1 + \lambda_t^0 K_t^T A_t^* \right)$$

$$\times \left[-(e_m)^2 + \left(1 - \eta_t \rho_t ||K_t^T + \lambda_t^0 A_t^*||^2 / \rho_t (1 + \lambda_t^0 K_t^T A_t^*) \right) e_t^2 \right]$$

$$\leq -\frac{\eta_t}{\rho_t} \left(1 + \lambda_t^0 K_t^T A_t^* \right) [-(e_m)^2 + \gamma e_t^2]$$

$$\leq -\frac{\eta_t}{\rho_t} \left(1 + \lambda_t^0 K_t^T A_t^* \right) [-(e_m)^2 + \gamma e_t^2] \leq 0. \quad \text{(41)}$$

Therefore, there must exist a positive constant $\gamma$ to meet the theoretical condition $1 - \pi_t > \gamma > 0$. In practical implementation, the dead zone range $e_m / \sqrt{\gamma}$ in (24) can be determined together via a small positive number (depending on the estimated equivalent disturbance level) or simply can use the variable dead zone parameter $e_m / (1 - \pi_t)^{1/2}$ to replace the fixed dead zone value $e_m / \sqrt{\gamma}$ in (24).

**Remark 2:** According to the classical deterministic system approach, there is no way for adaptive filtering to completely eliminate the estimation error, i.e., $e_t = 0$, rather than to minimize the estimation error and have the weight vector estimation $a_t$ be bounded not far from the optimal or sub-optimal vector $a^* [20] - [23], [28]. It should also be pointed
out that the bounded estimation error claim is fundamental to our approach to explain the link between kernel sparsification scheme and weight convergence via robust stability concept. This is because the former is to overcome the well-known overfitting problem by using a relatively smaller network to achieve a good generalization performance, i.e., a relatively smaller testing error rather than a training error. Since the testing error is unknown during the time of training, the sparsification scheme is based on the ALD condition, which looks for an approximation of the true/optimal or suboptimal sparsification scheme and weight convergence via robust stability concept. It is implemented by the growing method of taking the first input data as the initial kernel center, in an independent manner [7]. It is implemented by the growing method of taking the first input data as the initial kernel center, then the subsequent kernel center growing will be dependent on the ALD condition, which automatically decides both the network size \( m_t \) and the best possible kernel center approximation. Largely due to the model approximation, which is introduced in both the equivalent estimation error \( e_t \) and the recurrent weight term as in Proposition 1, it is no longer possible to ensure the weight convergence to its true value in a recurrent signal feedback system, as shown Fig. 1. Therefore, according to the robust stability theorem, we first use the Lyapunov function to prove that the weight is converged or bounded to the assumed optimal or suboptimal value against a bounded output disturbance, i.e., the \( L_2 \) robust stability. Then, we can extend it to the \( L_\infty \) robust stability condition with the exponentially weighted signal in a more practical case based on the conic sector theorem [25], [26].

B. Conic Sector Theorem and \( L_2 \)/\( L_\infty \) Analysis

In nonlinear system theory, the stability analysis can be established by the Lyapunov theory and the input–output theory [25], [26]. Among the two methods, the functional analysis based input–output theory provides a natural tool to answer questions about the robust stability of RNN systems. We only need to make minimal assumptions about the processes or systems, which we are investigating. In this paper, the conic sector theorem from the input–output analysis method is employed to analyze the robustness of RNN training algorithm. First, a brief review of the theorem is presented.

Consider the RNN feedback system in Fig. 1, which can be represented in an equivalent feedback system of Fig. 2 according to the conic sector theorem [25], [26] as follows:

\[
e_t = e_t - e_t^a = -H_2 e_t^a + e_t \tag{42}\]
\[
e_t^a = H_1 e_t \tag{43}\]
\[
r_t = H_2 e_t^a \tag{44}\]

where \( H_1, H_2 : L_{2e} \rightarrow L_2e \), and \( e_t, e_t^a, e_t^a \in L_{2e} \), where \( L_{2e} \) is the extended \( L_2 \) space in [25], \( e_t^a = \alpha_t K_t \) defined in (29) is called as the regression error vector [25], which is closely linked to the weight estimation error vector \( \alpha_t \).

The operator \( H_1 \) represents the RRKOL learning algorithm, and (42) is directly taken from the RRKOL error equation (28) with an operator \( H_2 = 1 \), i.e., \( r_t = e_t^a \) in the feedback loop of Fig. 2, because there is no linear controller and mismatch linear dynamics in the recurrent feedback RNN structure of Fig. 1, which is different from the linear control systems as in [25] and [26].

**Proposition 3:** Assume that \( e_t \) is bounded in \( L_2 \) and suppose that the following holds.

1) \( H_1 : e_t \rightarrow e_t^a \) satisfies

\[
\sum_{k=1}^{N} \left[ e_t^a \epsilon_t + \sigma e_t^2 / 2 \right] \geq -\vartheta. \tag{45}\]

2) \( H_2 : e_t \rightarrow r_t \) with \( e_t = e_t - r_t \) (temporarily assume \( r_t \neq e_t^a \) for a general case) satisfies

\[
\sum_{k=1}^{N} \left[ \sigma r_t^2 / 2 - r_t e_t^a \right] \leq -\omega \| r_t, e_t^a \|^2_N. \tag{46}\]

If conditions 1) and 2) hold for some positive constants \( \sigma, \vartheta, \omega \), then the feedback system (42)–(44) is stable in the sense of \( e_t^a, e_t \in L_2 \).

**Proof:** The following proof is slightly different from [25, Corollary 2.1] according to the specific feedback system in Fig. 2. From (42) and (45), we have

\[
\sum_{i=0}^{N} \left[ e_t^a \epsilon_t + \sigma e_t^2 / 2 \right] = \sum_{i=0}^{N} \left[ e_t^a \epsilon_t - r_t e_t^a + \sigma / 2 (e_t^2) - 2 e_t r_t + (r_t)^2 \right] \geq -\vartheta. \tag{47}\]

Combining (46) and (47)

\[
-\omega \| r_t, e_t^a \|^2_N + \sum_{i=0}^{N} \left[ e_t^a \epsilon_t - r_t e_t^a + \sigma / 2 (e_t^2) \right] \geq -\vartheta \tag{48}\]

and using the Schwartz inequality

\[
\omega \| r_t, e_t^a \|^2_N - \| e_t^a \|_N \| e_t \|_N - \| r_t \|_N \| e_t^a \|_N \leq \vartheta + \sigma / 2 \| e_t \|^2. \tag{49}\]

Assume that \( \| r_t, e_t^a \|^2_N \rightarrow \infty \) as \( N \rightarrow \infty \), therefore, according inequality (49), there has to be \( \omega \leq 0 \). This is a contradiction; therefore, \( \| r_t, e_t^a \|^2_N \) is bounded, i.e., \( r_t, e_t^a \in L_2 \), which leads to \( e_t \in L_2 \).

**Remark 3:** As mentioned before, we focus on the bounded nonzero disturbance to achieve robust stability of the RNN system based on the conic sector theory [25], [26]. The key point of the exponentially weighted adaptive process using the normalization factor \( p_t \) as defined in (23) is to relax the disturbance \( e_t \in L_2 \) in the bounded convergence proof of Proposition 2 and \( L_2 \) stability in Proposition 3, which requires \( e_t \) eventually to converge to zero to achieve...
a good result, i.e., a small $e_t$ and $e_t^\alpha$ [25], [26]. This is not practical, because the disturbance cannot be zero due to external and modeling disturbances. This problem can be solved by Propositions 4 and 5.

Due to the specific selection of the normalization factor in (23), the normalized error signals guarantee that the original signals $e_t$ and $e_t^\alpha$ are bounded according to the original operators $H_1$ and $H_2$ [25], [26]. Moreover, the same result can be extended to the normalized system as follows.

**Proposition 4:** The normalized signal $\tilde{e}_t = e_t / \sqrt{\rho_t} \in L_2$.

**Proof:** Summing both sides of (39) up to $N$ steps and dividing by $\eta_t (1 + \lambda^\alpha t^T A^2_t K^T t \bar{K} t A^T t)$, and assuming temporarily the steps with $\eta_t \neq 0$, since the system is stable in the case of $\eta_t = 0$, we have

$$\sum_{k=1}^{N} \left\{ e_t e_t^\alpha / \rho_t + |\bar{K} t A^2 t \bar{K} t A^T t| / 2 / \rho_t \right\} = \sum_{k=1}^{N} \left\{ \tilde{e}_t \tilde{e}_t^\alpha + \bar{\sigma} e_t^2 / 2 \right\} \geq -\Delta V$$

where $\Delta V = -\sum_{k=1}^{N} (|\bar{a}_t t^2 - |\bar{a}_t^2| / \eta_t \geq 0$, because for each $t$, the Lyapunov function (40) is guaranteed to be smaller or equal to zero. The normalized error signals are $\tilde{e}_t = e_t / \sqrt{\rho_t}$, $\tilde{e}_t^\alpha = e_t^\alpha / \sqrt{\rho_t}$, and the cone $\bar{\sigma} = \|K t A^2 t \bar{K} t A^T t\| / (1 + \lambda^\alpha t^T A^2 t A^T t) / \rho_t < 1$, which prevents the vanishing radius problem, i.e., $\bar{\sigma}$ is strictly smaller than one [25].

Therefore, both the original $e_t$ and the normalized $\tilde{e}_t$ with the normalization factor $\rho_t$ belong to $L_2$, which is in exactly the same format as in condition 1 (45) of Proposition 3.

Furthermore, since the operator $H_2 = 1$ and normalized $\tilde{H}_2 = 1$ are constants, the conic sector condition of $H_2$ is guaranteed [25], [26].

Following Proposition 4, since $\tilde{e}_t = e_t / \sqrt{\rho_t} \in L_2$, we have the following.

**Proposition 5:** If the equivalent disturbance $e_t \in L_\infty$, then we have $\tilde{e}_t$ and $\tilde{e}_t^\alpha \in L_\infty$.

**Proof:** Similar to both [25] and [26], we use the exponentially weighted signal, denoted by $\lambda^\beta = \beta x_t$, with $\beta > 1$ to show the $L_\infty$ extension. Given $\rho_t$ is defined in (23) and bounded away from zero, and since the normalized operator $\tilde{H}_2^\beta = \beta \tilde{H}_2 \beta^{-1}$ is strictly inside the cone (consider this particular system with $H_2 = \bar{H}_2 = 1$), then according to Proposition 4 that $L_2$ stability is ensured with the sector stability theorem, i.e., Proposition 3. Therefore, there exists a scalar $K < \infty$, such that the error feedback system in Fig. 2 with the normalization factor $\rho_t$ has the following:

$$\| (e_t^\alpha)^\beta \|_N \leq K \| (\tilde{e}_t)^\beta \|_N \forall N \geq 0.$$ (51)

Notice that

$$\| (\tilde{e}_t)^\beta \|_N \geq (\beta^N e_t^\alpha_N)^2$$ (52)

and

$$\| (\tilde{e}_t^\alpha)^\beta \|_N \leq \| (\tilde{e}_t)^\beta \|_N \sum_{i=0}^{N} \beta^{2i} \leq \| (\tilde{e}_t)^\beta \|_N \frac{\beta^2}{1 - \beta^{-2}}$$ (53)

since $\beta > 1$. Combining (51)–(53), we can conclude that uniformly in $N$

$$\langle e_t^\alpha \rangle^2 \leq \| (\tilde{e}_t)^\beta \|_N \frac{K}{\rho (1 - \beta^{-2})}$$ (54)

where

$$\rho \triangleq \min \rho_t > 0.$$ (55)

Since the normalized feedback system in Fig. 2 is stable, Propositions 3 and 4, similarly as in [25] and [26] for $\rho_t \in L_\infty$ and $\beta^2 = \mu - 1$, we can conclude that $e_t \in L_2 \Rightarrow \tilde{e}_t \in L_2$, and consequently, $e_t^\alpha \in L_2$. Hence, from inequality (54), we can conclude the condition that ensures the boundedness of $e_t^\alpha$

$$1 - 2 \| (\tilde{e}_t)^\beta \|_N \frac{K}{\rho (1 - \mu)} \geq \mu$$ (56)

where $\gamma_2$ is defined as the $L_2$ gain parameter of the estimate as in [25] and [26], which can be rewritten as

$$(1 - \mu)^2 > 2 \| (\tilde{e}_t)^\beta \|_N \frac{1}{\rho (\gamma_2)^2}.$$ (57)

Since all the items in the numerator of the right-hand side of (57) are bounded and $\mu$ ranges $(0, 1)$ (open set), there exists a $\tilde{\rho}$ that makes (57) true. This completes the proof. ■

**Remark 4:** Only $e_t^\alpha \in L_\infty$ is required in Proposition 5. As we defined in (30), that $e_t$ has three components. The last one $e_t^\alpha$ is an external disturbance as in the RNN feedback structure (Fig. 1), and we cannot control or reduce it according to the input–output stability theory [25], [26]. However, the effect of the modeling error, i.e., the first two terms of $e_t$, can be reduced through the coherence or ALD-related kernel center selection and sparsification criterion (5) as used in the RRKOL algorithm. Moreover, it is impossible in practical case that the equivalent disturbance will eventually go to zero, i.e., $e_t \in L_2$ as required by Propositions 2 and 3. Therefore, Proposition 5 gives a better result by relaxing it to $e_t \in L_\infty$, which implies that model (8) can be optimal or suboptimal and allow a nonzero equivalent disturbance $e_t$, which includes both the external disturbance $e_t^\alpha$ and the modeling error, i.e., the first two terms of $e_t$ as in (30), to obtain the robust stability of the RRKOL algorithm. This is to achieve a tighter bound of the conic sector condition $\bar{\sigma} < 1$ [defined (50)] and a relatively small regression error $e_t^\alpha = \bar{a}_t K t$ as defined in (29), i.e., a relatively small weight estimation error vector $\bar{a}_t$ under the nonzero disturbance as a result of the optimal or suboptimal kernel center selection procedure, i.e., the first two terms of $e_t$ defined in (30) will be gradually eliminated through the coherence criterion (5). This explains why our robust approach of the RRKOL algorithm has a strong robustness property against the disturbance and model uncertainty. More importantly, as pointed out before, the ALD-related sparsification allows an approximation of the optimal or suboptimal selection of the kernel center and the number of kernel functions with a growing method and limited kernel number. As a result of combination of the $L_\infty$ robust stability and ALD-related sparsification, the RRKOL can take a nonzero disturbance $e_t$ to obtain $e_t^\alpha \in L_\infty$ and, in turn, a reasonable small $e_t$ in
the sense of $L_\infty$ stability. Note that the reasonable small $e_t$ is referred to the minimization of the disturbance $e_t$ in (30), since the first two items can be eliminated through the coherence sparsification (7).

We achieve a tradeoff between weight convergence and robust stability of the RRKOL algorithm in the sense that an optimal or suboptimal solution can be guaranteed depending on the variable length of the kernel model (4) to approximate the ideal kernel model (8). As the estimated model (4) grows, the estimated weights will eventually approximate the ones in ideal model (8) under noise free condition with unlimited input data. However, due to disturbances and inherent model uncertainty, for example, the equivalent disturbance defined in (30), there is always an estimation error. In particular, we need to consider the fact that Proposition 2 only proves a bounded weight error norm. Therefore, we should emphasize importance of the introduction of conic sector-based stability analysis with the specific normalized adaptive RRKOL algorithm in Propositions 3 and 5. This guarantees the recurrent $L_\infty$ stability in term that the equivalent disturbance in the proposed error feedback system (see Fig. 2) may not necessarily approximate to zero. This also implies that the system will still be stable in the sense of $L_\infty$ even in case that the estimated weight and the size of the network may not approximate the optimal or ideal values as in (8). This is exactly meaning of the optimal or suboptimal solutions in term of the stability and weight convergence analysis. Incidentally, there is, if it is not unexpected, a match between generalization performance and stability analysis of the RRKOL algorithm. As well known, the generalization performance of neural networks is always a compromise to the weight convergence in term of testing error. We do not need the estimated model (4) to approximate exactly to a specific ideal or optimal model (8), because the optimal model is based on existing best data selection, i.e., assuming that the best available input data for the ideal kernel centers are already available as defined in (8). This may not necessarily be the case for unknown future testing data in the online learning case. Therefore, we believe that our research agrees with [31] and [32], in certain extent, for example, to use the linear approximation in kernel space instead of the projector concept in [31], such that the best weight with convergence may not necessarily match the sparsification criterions (6). Therefore, the ideal model (8) may have an optimal dimension $m^*$ with ideal weight, which is different from the estimated one in model (4). The latter may emphasize the generalization performance through a small network rather than a large network, which takes care of more on the weight convergence. This is why we need to introduce the sparsification scheme. It uses a general small size network to avoid overfitting as in [20], i.e., the model size $m_t$ in (4). It may or may not equal to the optimal model size $m^*$ in (8) to improve the generalization performance. The RRKOL may have a nonzero estimation error, while the system stability is guaranteed as proved in Proposition 5 in Section III for a tradeoff between accuracy of the weight convergence and generalization performance. Note also that this tradeoff is important for time-vary system, in particular, like the simulation B in Section IV, because there is no fixed model can be an optimal model rather than a suboptimal one.

IV. SIMULATIONS

In this section, we experimentally test the performance of the RRKOL algorithm on three benchmark problems and one real-time data set for the time-series prediction. The performance comparisons of the RRKOL algorithm with the two most typical kernel online learning algorithms (KNLMS [7] and QKLMS [5]) are presented. The learning accuracy is quantified using the root-mean-square error (RMSE), which is defined as $\text{RMSE} = (\sum_{t=1}^{n} (y_t^* - y_t)^2 / n)^{1/2}$. We have carefully tuned the parameters for all selected algorithms, and the optimal parameter values are listed in Tables I and II.

A. Nonlinear System

In the first experiment, we analyzed the nonlinear system described by the differential equation

$$y_t^* = (0.8 - 0.5\exp(-y_{t-1}^2))y_{t-1}^*$$
$$- (0.3 + 0.9\exp(-y_{t-1}^2))y_{t-2}^* + 0.1\sin(y_{t-1}^* \pi)$$

where $y_t^*$ is the desired output. This highly nonlinear time series has been investigated in [7]. The data were generated using an initial condition (0.1, 0.1); a data set with 1000 samples was used for training, and the following 100 samples were used as testing data. A recurrent depth of 10 was used for the feedback output $y_t$ together with two delayed system input patterns $u_{t-1} = y_{t-1}^*$ and $u_{t-2} = y_{t-2}^*$, i.e., $x_t = [u_{t-1}, u_{t-2}, y_{t-1}, \ldots, y_{t-10}]^T$. Note that $y_t^*$ is the measurement of the real-time series from equation (58) in this simulation, five delayed values of $y_t^*$ are also used for inputs of the feedforward kernel algorithms QKLMS and KNLMS for comparison purpose, because they do not use the delayed recurrent feedback as inputs (similar concepts are used for the next two examples in Sections IV-B and IV-C as well). The preliminary experiments we conducted lead us to adopt the parameter settings given in Table II, where the same notation applied in [5] and [7] is used. To test the statistical robustness of RRKOL, a white Gaussian noise with variance 0.01 is added in the data for the comparison of the three algorithms. See the results over 1000 Monte Carlo runs in Table II. Table II also gives the CPU running time for each simulation, the final dictionary size, and the testing RMSE.

The RRKOL algorithm has the best performance as the prediction data match the desired data very well as summarized in Table II. Fig. 3 compares the RMSE values for each of the three methods at each training step. Fig. 4 shows the trace of the $L_2$ norms of the weights at each step. The dictionary size of the RRKOL algorithm at each training step is shown in Fig. 5. It is interesting to note that one of the most significant parameter selections of the RRKOL algorithm is the value of $\hat{\varepsilon}_m$ in (24) [we use the variable dead zone value $\hat{\varepsilon}_m/(1 - \pi_1)^{1/2}$ as explained (40)], because it represents the estimation disturbance level of $\hat{\varepsilon}_t$, as discussed in Section III. Note that Table I shows the performance results of the algorithm using
different values for the certain parameters. It is clear that the RRKOL algorithm is relatively robust against the disturbance for a reasonable range of \( \varepsilon_m \). Furthermore, due to the ON–OFF property of the adaptive learning rate \( \eta_t \) in (24), we are allowed to select a relative larger constant parameter \( \eta \), and in turn, a smaller RMSE in learning curve of the RRKOL algorithm. The other parameter values, such as \( \lambda_0, \nu, \) and \( \bar{\rho} \), are relatively insensitive, because they function as the initial values of the adaptive system. Finally, to test the robustness of the RRKOL algorithm under different recurrent delay steps, Fig. 6 shows the RMSE comparison curves. We can see that a reasonably larger \( d \) leads to a relatively smaller RMSE value as the use of recurrent delay signal is one of the main advantages of the RRKOL algorithm.

### B. Wind Data Prediction

The wind data consists of three dynamic regions: 1) low; 2) mediate; and 3) high regions, and in each region, there are two directions: the east and north directions.\(^1\) The wind data are highly intermittent and nonstationary [30]. The original data set in this experiment is from the wind data recorded from the low dynamics region and the north direction. The training data consist of 2000 samples, and the testing data involve the subsequent 500 samples. The prediction problem uses the previous ten data points to predict the present one; therefore, the input vector can be formulated as \( x_t = [y_{t-1}, \ldots, y_{t-10}] \). The first ten data points were used as input data, and therefore, the training was started from the eleventh data point. One-step prediction was performed for the wind time-series prediction.

To test the statistical robustness of RRKOL, a white Gaussian noise with variance 0.01 is added in the data for the comparison of the three algorithms. See the results over 1000 Monte Carlo runs in Table II. The parameter settings displayed in Table II are used, and the same notations applied in [5] and [7] are used.

Fig. 7 compares the RMSE at each training step by using the three kernel online learning algorithms. Fig. 8 shows the trace of \( L_2 \) norms of the weights at each step. The percentages

\(^1\)Available at: http://www.commsp.ee.ic.ac.uk/~mandic/research/Smart-Grid-and-Renewables.htm
of nonzero $\eta_t$ over the 2000 training step is 0.2632. The dictionary size of the RRKOL algorithm at each training step is shown in Fig. 9. In summary, the RRKOL algorithm achieves a relatively smoother learning curve (in terms of the generalization performance) and better weight convergence and robust stability compared with the feedforward network-based training of the two other kernel algorithms, as shown in Figs. 7 and 8.

C. Mackey–Glass Time-Series Prediction

The Mackey–Glass equation is a chaotic time-series generator. It is widely recognized as one of the benchmark time-series problems, generated from the following time-delay ordinary differential equation:

$$\frac{dx(t)}{dt} = -\beta x(t) + \frac{\alpha x(t - \tau)}{1 + x(t - \tau)^5}$$  \hspace{1cm} (59)

where $\beta = 0.1$, $\alpha = 0.2$, and $\tau = 30$. The time series was discretized using a sampling period of 6 s based on the Runge–Kutta method, which generates the ideal system output $y^*_t$. The prediction problem uses the previous ten data points to predict the present one; therefore, the input vector can be formulated as $x_t = [y_{t-1}, \ldots, y_{t-10}]$. A segment of 2000 samples was used as training data, and the subsequent 200 points were used as testing data. The first ten data points were used as input data, and therefore, the training was started from the eleventh data point. One-step prediction was performed for the chaotic time series.

To test the statistical robustness of RRKOL, a white Gaussian noise with variance 0.01 is added in the data for the comparison of the three algorithms. Table II shows the results over 1000 Monte Carlo runs. The parameter settings displayed in Table II were used, where the notation is the same as in [1] and [7].

![Fig. 9. Dictionary size of the RRKOL algorithm at each training step for wind data.](image)

![Fig. 10. Learning curve for the KNMLS, QKLMS, and RRKOL algorithms applied to the testing data of the Mackey–Glass time series.](image)

![Fig. 11. Full trace of the $L_2$ norms of the weights for the Mackey–Glass time series.](image)

![Fig. 12. Dictionary size of the RRKOL algorithm at each training step for the Mackey–Glass time series.](image)

FIG. 9. Dictionary size of the RRKOL algorithm at each training step for wind data.

FIG. 10. Learning curve for the KNMLS, QKLMS, and RRKOL algorithms applied to the testing data of the Mackey–Glass time series.

FIG. 11. Full trace of the $L_2$ norms of the weights for the Mackey–Glass time series.

FIG. 12. Dictionary size of the RRKOL algorithm at each training step for the Mackey–Glass time series.

Fig. 10 compares the RMSE values of the three training methods at each training step. Fig. 11 shows the trace of the $L_2$ norms of the weights at each step. The fraction of the 2000 training steps for which $\eta_t = \eta$, i.e., the nonzero adaptive learning rate, is 0.8720. The dictionary size of the RRKOL algorithm at each training step is shown in Fig. 12. It is interesting to note that according to the no free lunch theory, the RRKOL algorithm, like many other adaptive kernel filtering algorithms, may not always be the best for some model-dependent problems. The QKLMS algorithm achieves the best generalization performance in Fig. 10, even though the RRKOL algorithm has a relatively better weight convergence.
and robust stability property compared with the QKLMS algorithm, as shown in Fig. 11.

V. CONCLUSION

We proposed a new class of kernel learning algorithms called RRKOL. This novel type of algorithm combines the advantages of both kernel and recurrent learning methods. A structural risk minimization cost function that includes a recurrent learning feedback term was developed. Furthermore, the weight convergence and robust stability condition was proved and connected to the kernel sparsification scheme using an integrated approach such that the RRKOL algorithm automatically weighs the regularized term in the full derivative (16), so that we can minimize not only the estimation error $e_t$ but also improve generalization performance via the sparsification.

REFERENCES


Qing Song (M’92–SM'11) received the B.S. degree from the Harbin Shipbuilding Engineering Institute, Harbin, China, in 1982, the M.S. degree from Dalian Maritime University, Dalian, China, in 1986, and the Ph.D. degree from the Industrial Control Centre, Strathclyde University, Glasgow, U.K., in 1992. He was an Assistant Professor with Dalian Maritime University from 1986 to 1987. He was a Research Engineer with System and Control Pte Ltd. He joined the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, as a Lecturer in 1992, where he is currently an Associate Professor. He held short visiting research positions with Johns Hopkins University, Baltimore, MD, USA, and Tsinghua University, Beijing, China, respectively. He is an Active Industrial Engineer with Xyratex Technologies, a Leading Manufacturer of Photolithography Systems, Nanyang Technological University, Singapore, as an Academic Researcher. He has published many refereed international journal and conference publications, which are focused in the areas of neural networks and artificial intelligence (www.ntu.edu.sg/home/eqsong).
Xu Zhao received the B.Eng. and M.Eng. degrees in automatic control from Northwestern Polytechnical University, Xi’an, China, in 2005 and 2008, respectively, and the Ph.D. degree in control and instrumentation from Nanyang Technological University, Singapore, in 2013. Her current research interests include data analytics, control, and neural networks.

Haijin Fan received the Ph.D. degree in control and instrumentation from Nanyang Technological University, Singapore, in 2013. He was with the Institute of High Performance Computing, Singapore, after his Ph.D. studies.

Danwei Wang received the B.E. degree from the South China University of Technology, China, in 1982, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 1984 and 1989, respectively. He is currently a Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore. He is the Director of the Centre for System Intelligence and Efficiency and the Director of the STE–NTU Corporation Laboratory, NTU. He has published widely in the areas of iterative learning control, repetitive control, fault diagnosis and failure prognosis, satellite formation dynamics and control, and manipulator/mobile robot dynamics, path planning, and control. Dr. Wang was a recipient of Alexander von Humboldt Fellowship, Germany. He has served as a General Chairman, Technical Chairman, and various positions in international conferences. He has served as an Associate Editor of the Conference Editorial Board of the IEEE Control Systems Society. He is an Associate Editor of the International Journal of Humanoid Robotics and an Invited Guest Editor of various international journals.