IEEE MELECON 2004, May 12-15, 2004, Dubrovnik, Croatia

Complex-Valued Estimation of Wind Profile and Wind Power

Su Lee Goh, D.H.Popović and D.P.Mandic
Department of Electrical and Electronic Engineering, Imperial College, London SW7 2BT, United Kingdom
{su.goh, dp, d.mandic}@imperial.ac.uk

Abstract— **This paper addresses the problem of wind profile and wind turbine power estimation. A complex-valued pipelined recurrent neural network (CPRNN) architecture is proposed. The network is trained by the complex-valued real-time recurrent learning (CRTRL) algoritm with a general 'fully' complex activation function which makes it suitable for forecasting wind signal in its complex form (speed and direction). The subsequent complex-value based prediction of wind turbine power is shown to significantly differ from the one based on independent prediction of wind speed and wind direction with the latter mainly being more optimistic in predicting the turbine power output.**

I. Introduction

The last few years have witnessed a dramatic increase in the demand for wind power. Advances in wind turbine technology and rich and comparatively cheap wind resources improve prospects of deploying the wind power in daily power system operation as both the base generation and reserve power supply for peak periods. As a result, wind turbine (WT) power station operators aim at getting the most from their power generation process. It implies an increased need not only for accurate and reliable information on usually unpredictable wind dynamics but also for accurate estimation of the corresponding changes in the wind turbine power output. This paper will focus on developing a new method and novel design of the neural network based forecasting system to be used for estimation of wind turbine power output. For the first time, a turbine power forecast will be based on a vector-field wind signal forecast rather than on wind speed and wind direction being forecasted separately.

The power generated by wind turbines changes because of the continous fluctuation of both the wind speed and its direction. Various field measurements have shown that direction of wind as compared with wind speed has less influence on WT power output because each turbine is usually built to face into the wind when operating. Consequently, and especially at stronger winds, there is no significant difference in the power generated for different wind directions. However, the impact of wind direction on power output is more prominent at milder winds since they usually come from much wider directions [1]. The importance of wind direction is of further significance in spatial correlation studies which aim to assess the influence of WT position in a wind park [2]. All this emphasises the need to process a wind signal as a vector field, i.e., in the complex plane defined by wind speed and its direction.

Since the wind data are highly nonlinear and nonstationary, the identification of parameters and contributing factors to describe the power supplied by this unpredictable, non-controllable and intermittent source is not trivial. Classical parametric methods such as time-series (AR, ARMA) methods used widely in short and long-term forecasting have limitations in dealing with the nonlinear and nonstationary nature of wind signals and are therefore prone to numerical instability and inaccuracy. The forecasting methods based on a neural network (NN) approach have been shown as the most promising in terms of forecasting accuracy and efficient computation because of their approximation ability for nonlinear mappings and generalization in a non-parametric fashion [3]. In particular, more recently introduced recurrent neural networks (RNNs) as nonlinear dynamical systems which possess both short– and long– term memory (due to their feedback) and exhibit attractor dynamics are shown as particularly suitable for prediction of nonlinear and nonstationary signals [4].

Most methods for short and long term prediction of complex time series consider them as two bivariate independent real time series instead as components on one complex variable. A pair of activation function is employed to separately process real and imaginary components of the weighted sum of the input signal. This way, the output from the complex activation function takes split paths of two unrelated real valued gradients [5].

In this paper, a pipelined recurrent neural network (PRNN) architecture trained by the complex-valued real-time recurrent learning (CRTRL) algoritm with a general 'fully' complex activation function is proposed for estimation of wind profile and wind turbine power. A complex nature of wind dynamics is discussed in Sec II. In Sec III, main features of the complex-valued estimation framework based on complex-valued pipelined recurrent neural network (CPRNN) trained by the CRTRL algorithm are described. Simulation results are presented in Sec IV.

II. Wind Characteristics and Wind Power

The power generated by a wind turbine is inherently dependent on the wind speed. More specifically, the power which can be extracted from the airflow is given by

$$P_w = \frac{\rho}{2} C_p(\alpha, \theta) A w^3 \quad [W] \tag{1}$$

where ρ is the air density [kg/m^3], C_p is the performance or power coefficient, α is tip speed ratio, that is, the ratio between the blade tip speed w_t and the wind speed upstream the rotor w [m/s], θ is the blade pitch angle and A is the area swept by the rotor [m^2].

The relation between wind speed and generated power is usually given by the power curve of the wind turbine. A power curve, however, is derived under a set of assumptions regarding the wind speed and air density. These so-called 'ideal data' are often impractical in estimating the actual power output of each wind turbine due to a turbine distance and relative position in the wind park with respect to the meteorological tower(s). The impact of wind direction is also notable, especially during mild winds when the angle of wind direction can sweep a very wide range influencing significantly the power output [1]. Further, the topographic conditions often make the power generated by different turbines be very different, even under equivalent weather conditions [2]. The introduction of an extra parameter k_d in the power model given by (1) to reflect the relationship between the turbine position and reference anemometar is not sufficient to account for unpredictability in wind direction. More specifically, the changing wind conditions would require k_d to be a wind-profile and time dependent function rather than a single parameter.

As a result, the estimation of actual wind turbine power should be based on processing the wind-vector measurements (speed, pressure, temperature and direction). In particular, wind speed and wind direction can be used as two separate but strongly correlated inputs which define a wind signal (Fig 1). The forecasting tool will then be able not only to reflect the influence of wind speed on wind power output but also to capture the influence of wind direction at certain ranges of wind direction angle.

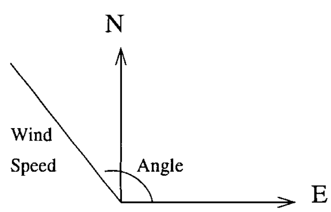

Fig. 1. Wind data as a complex [speed,direction] quantity

III. The Complex-Valued Estimation Framework

Current research on processing signals in the complex domain has mainly been based on a split-complex activation function (AF). In a split-complex AF, the real and imaginary components of the input signal x are split and fed through the real-valued activation function $f_R(x) = f_I(x)$, $x \in \mathbb{R}$. The equation of the split activation function is given as $f(x) = f_R(Re(x)) + jf_I(Im(x))$. Although bounded, a split-complex AF cannot be analytic, and thus cannot cater for signals with strong correlation between magnitude and phase [5]. The newly proposed complex-valued real time recurrent learning (CRTRL) algorithm for a fully connected recurrent neural network (FCRNN) is however suitable for adaptive filtering of complex-valued nonlinear and nonstationary signals with strong component correlations. In addition, this algorithm is generic and represents a natural extension of the real-valued RTRL.

A. The basic algorithm

Fig 2 shows a FCRNN, which consist of N neurons with p external inputs. The network has two distinct layers consisting of the external input-feedback layer and a layer of processing elements. Let $y_l(k)$ denote the complex-valued output of each neuron, $l = 1, \ldots, N$ at time index k and $\mathbf{s}(k)$ the $(1 \times p)$ external complex-valued input wind signal vector. The overall input to the network $\mathbf{Z}(k)$ represents a concatenation of vectors $\mathbf{y}(k)$, $\mathbf{s}(k)$ and the

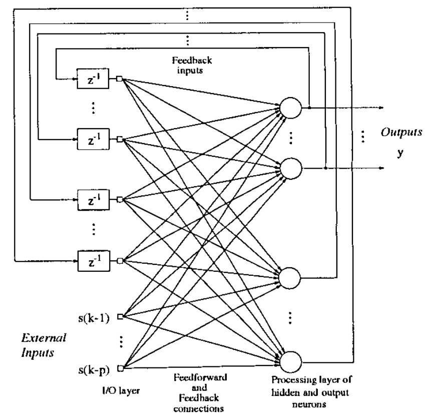

Fig. 2. A fully connected recurrent neural network for prediction

bias input $(1 + j)$, and is given by

$$\begin{aligned}\mathbf{Z}(k) &= [s(k-1), \ldots, s(k-p), 1+j, y_1(k-1), \\ &\quad \ldots, y_N(k-1)]^T \\ &= Z_n^r(k) + jZ_n^i(k), \quad n = 1, \ldots, p+N+1 \qquad (2)\end{aligned}$$

where $j = \sqrt{-1}$ and the superscripts $(\cdot)^r$ and $(\cdot)^i$ denote respectively the real and imaginary part of a complex number. A complex-valued weight matrix for the network is denoted by $\mathbf{W}$, where for the nth neuron, its weights form a $(p + N + 1) \times 1$ dimensional weight vector $\mathbf{w}_l^T = [w_{l,1}, \ldots, w_{l,p+N+1}]$. The output of each neuron can be expressed as $y_l(k) = \Phi(net_l(k)), \quad l = 1, \ldots, N$, where

$$net_l(k) = \sum_{n=1}^{p+N+1} w_{l,n}(k) Z_n(k) \qquad (3)$$

is the net input to lth node at time index k. For simplicity, state that

$$y_l(k) = \Phi^r(net_l(k)) + j\Phi^i(net_l(k)) = u_l(k) + jv_l(k) \qquad (4)$$

where Φ is a complex nonlinear activation function of a neuron.

B. Complex-valued real time recurrent learning (CRTRL) algorithm

The output error which consists of its real and imaginary parts can be expressed as

$$e(k) = d(k) - y_1(k) = e^r(k) + je^i(k) \qquad (5)$$

$$e^r(k) = d^r(k) - u_1(k), \quad e^i(k) = d^i(k) - v_1(k) \qquad (6)$$

where $d(k)$ is the teaching signal. For real-time applications and gradient descent algorithms the cost function of the network is given by $E(k) = \frac{1}{2}|e(k)|^2 = \frac{1}{2}e(k)e^*(k) = \frac{1}{2}\left[(e^r)^2 + (e^i)^2\right]$ [6], where $(\cdot)^*$ denotes the complex conjugate. The CRTRL aims at minimising the error by recursively altering the weight coefficients based on the gradient search technique, given by

$$w_{l,n}(k+1) = w_{l,n}(k) - \eta \nabla_{w_{l,n}} E(k)_{|w_{l,n}=w_{l,n}(k)} \qquad (7)$$

Notice that $E(k)$ is a real-valued function and to calculate the gradient, the partial derivates of $E(k)$ with respect to the real and imaginary part of the weight coefficients separately have to be derived, as

$$\nabla_{w_{l,n}} E(k) = \frac{\partial E(k)}{\partial w_{l,n}^r} + j\frac{\partial E(k)}{\partial w_{l,n}^i} \qquad (8)$$

Calculating the gradient of the cost function with respect to the real part of the complex weight gives[1]

$$\frac{\partial E(k)}{\partial w^r_{l,n}(k)} = \frac{\partial E}{\partial u_1}\left(\frac{\partial u_1(k)}{\partial w^r_{l,n}(k)}\right) + \frac{\partial E}{\partial v_1}\left(\frac{\partial v_1(k)}{\partial w^r_{l,n}(k)}\right) \quad (9)$$

Similarly, the derivative of the cost function with respect to the imaginary part of the complex weight yields

$$\frac{\partial E(k)}{\partial w^i_{l,n}(k)} = \frac{\partial E}{\partial u_1}\left(\frac{\partial u_1(k)}{\partial w^i_{l,n}(k)}\right) + \frac{\partial E}{\partial v_1}\left(\frac{\partial v_1(k)}{\partial w^i_{l,n}(k)}\right), \quad (10)$$

The factors $\frac{\partial u_1(k)}{\partial w^r_{l,n}(k)}$, $\frac{\partial v_1(k)}{\partial w^r_{l,n}(k)}$, $\frac{\partial u_1(k)}{\partial w^i_{l,n}(k)}$ and $\frac{\partial v_1(k)}{\partial w^i_{l,n}(k)}$ are measures of sensitivity of the output of the lth neuron at time k to a small variation in the value of $w_{l,n}(k)$. For convenience, denote the corresponding sensitivities as $\pi^{rr}(k) = \frac{\partial u_1(k)}{\partial w^r_{l,n}(k)}$, $\pi^{ir}(k) = \frac{\partial v_1(k)}{\partial w^r_{l,n}(k)}$, $\pi^{ri}(k) = \frac{\partial u_1(k)}{\partial w^i_{l,n}(k)}$ and $\pi^{ii}(k) = \frac{\partial v_1(k)}{\partial w^i_{l,n}(k)}$. For a complex function to be analytic at a point in $\mathbb{C}$, it needs to satisfy the Cauchy-Riemann[2] equations [5]. To arrive at the Cauchy-Riemann equations, the partial derivatives (sensitivities) along the real and imaginary axes should be equal, that is $\pi(k) = \pi^{rr}(k) + j\pi^{ir}(k) = \pi^{ii}(k) - j\pi^{ri}(k)$. Equating the real and imaginary parts of the sensitivies leads to

$$\pi^{rr}(k) = \pi^{ii}(k), \pi^{ri}(k) = -\pi^{ir}(k) \quad (11)$$

By using the Cauchy-Riemann equations, a more compact representation of $\nabla_{\mathbf{w}}E(k)$ is obtained as $\nabla_{\mathbf{W}}E(k) \equiv e(k)\pi^*(k)$. The total weight matrix update is then

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \eta e(k)\pi^*(k) \quad (12)$$

with the initial condition $\pi(0) = \mathbf{0}$. Following the approach from [7], the update for the sensitivities $\pi^*(k) = \pi^{rr}(k) + j\pi^{ri}(k)$ can be derived as

$$\pi^*(k) = \{\Phi^*(k)\}' \left[\delta_{ln} Z^*_n(k) + \mathbf{w}^*_1(k)\pi^*(k-1)\right] \quad (13)$$

where

$$\delta_{ln} = \begin{cases} 1 & ,l = n \\ 0 & ,l \neq n \end{cases} \quad (14)$$

is the Kronecker delta.

C. The complex-valued pipelined recurrent neural network (CPRNN)

The CPRNN architecture contains M modules of FCRNNs connected in a nested manner as shown in Fig 3. The $(p \times 1)$ dimensional external complex-valued wind signal vector $\mathbf{s}^T(n) = [s(n-1), \ldots, s(n-p)]$ is delayed by m time steps $(z^{-m}\mathbf{I})$ before feeding the module m, where z^{-m}, $m = 1, \ldots, M$ denotes the m-step time delay operator, and $\mathbf{I}$ is the $(p \times p)$ dimensional identity matrix. The complex-valued weight vectors $\mathbf{w}_l$ are embodied in an $(p+N+1) \times N$ dimensional weight matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_N]$. All the modules operate using the same weight matrix $\mathbf{W}$. A full mathematical description of the CPRNN is given by [8]

$$y_{t,l}(k) = \Phi(net_{t,l}(k)), \quad t = 1, 2, \ldots, M \quad (15)$$

[1]We derive the CRTRL for prediction applications (only one output y_1), however the derivation is general enough to be straight forwardly extended to a RNN with more than one output.

[2]Cauchy-Riemann equations state that the partial derivatives of a function $f(z) = u(x,y) + jv(x,y)$ along the real and imaginary axes should be equal: $f'(z) = \frac{\partial u}{\partial x} + j\frac{\partial v}{\partial x} = \frac{\partial v}{\partial y} - j\frac{\partial u}{\partial y}$. Therefore, the Cauchy-Riemann equations are as: $\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}$.

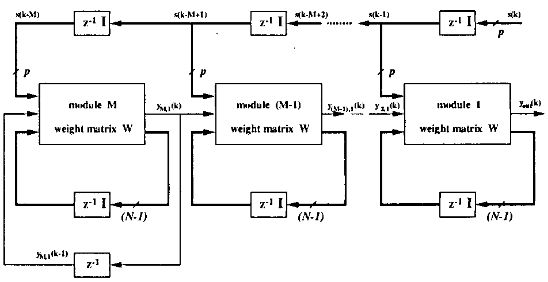

Fig. 3. Pipelined Recurrent Neural Network (PRNN)

$$net_{t,l}(k) = \sum_{n=1}^{p+N+1} w_{l,n}(k) Z_{t,n}(k), \quad \begin{array}{l} l = 1, \ldots, N \\ n = 1, \ldots, p+N+1 \end{array} \quad (16)$$

$$\begin{aligned} \mathbf{Z}^T_t(k) &= [s(k-t), \ldots, s(k-t-p+1), 1, \\ &\quad y_{t+1,1}(k-1), y_{t,2}(k-1), \ldots, y_{t,N}(k-1)] \\ &\quad for \quad 1 \leq t \leq M-1 \end{aligned} \quad (17)$$

$$\begin{aligned} \mathbf{Z}^T_M(k) &= [s(k-M), \ldots, s(k-M-p+1), 1, \\ &\quad y_{M,1}(k-1), y_{M,2}(k-1), \ldots, y_{M,N}(k-1)] \\ &\quad for \quad t = M \end{aligned} \quad (18)$$

where $\Phi(\cdot)$ is the activation function. For simplicity, state that

$$\begin{aligned} y_{t,l}(k) &= \Phi^r(net_{t,l}(k)) + j\Phi^i(net_{t,l}(k)) \\ &= u_{t,l}(k) + jv_{t,l}(k) \end{aligned} \quad (19)$$

The overall output signal of the CPRNN is $y_{1,1}(k)$, the output of the first neuron of the first module. At every time step k, for every module $t, t = 1, \ldots, M$, the one-step forward prediction error $e_t(k)$ associated with a module is then defined as $e_t(k) = s(k-t+1) - y_{t,1}(k) = e^r_t(k) + je^i_t(k)$. Since the CPRNN consists of M modules, a total of M forward prediction error signals are calculated. The original cost function introduced in [7] is modified to suit the complex domain as

$$E(k) = \sum_{t=1}^{M} \lambda(k)\,|e_t(k)|^2 = \sum_{t=1}^{M} \lambda(k)\,[e_t(k)e^*_t(k)] \quad (20)$$

which is the weighted sum of squared errors at the output of every module of the CPRNN and $\lambda(k)$ is a possible variable forgetting factor. The aim is to minimise (20) along the entire CPRNN. Hence, the weight correction for the nth weight of neuron l at the time instant k is derived as $\Delta w_{l,n}(k) = -\eta \frac{\partial}{\partial w_{l,n}(k)}\left(\sum_{t=1}^{M} \lambda(k)\,|e_t(k)|^2\right)$. The weight update of the CPRNN is finally given by

$$\begin{aligned} w_{l,n}(k+1) &= w_{l,n}(k) - \eta\nabla_{w_{l,n}} E(k)_{|w_{l,n}=w_{l,n}(k)} \\ &= w_{l,n}(k) + \eta e_t(k)\pi^*_t(k) \\ &= w_{l,n}(k) + \eta e_t(k)\{\Phi^*(net_{t,l}(k))\}' \times \\ &\quad [\delta_{ln} Z^*_{t,n}(k) + \mathbf{w}^*_1(k)\pi^*_t(k-1)] \end{aligned} \quad (21)$$

IV. Simulation Results

The complex-valued real-world wind signal is used as an input to the CPRNN. The wind data are obtained from the website *'http://mesonet.agron.iastate.edu/request/awos/1min.php'* which gives wind velocities and directions at 1-min averages.

The nonlinearity at the neuron was chosen to be the logistic sigmoid function $\Phi(x) = \frac{1}{1+e^{-\beta x}}$ where x is complex-valued. The slope was chosen to be $\beta = 1$ and learning rate for the CPRNN architecture was choosen to be $\eta = 0.01$. The forgetting factor for the CPRNN architecture was $\lambda = 0.8$. The number of modules is chosen to be $M = 5$, number of neurons in a module $N = 2$ and number of tap inputs $p = 5$.

Fig 4 shows the prediction performance of the CPRNN applied to the complex-valued wind signal. It can be observed that the CPRNN was able to track the complex wind signal very accurately. On the contrary, Fig 5 shows a poor wind prediction when algorithm with split-complex activation function is employed.

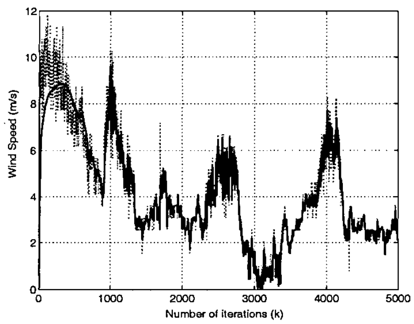

Fig. 4. Prediction of complex wind signal using 'fully' CRTRL. Solid curve: nonlinear prediction of wind signal. Dashed curve: actual wind signal

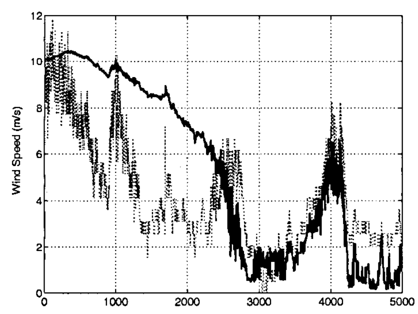

Fig. 5. Prediction of complex wind signal using CRTRL employing split activation function. Solid curve: nonlinear prediction of wind signal. Dashed curve: actual wind signal

To illustrate the importance of wind direction, the power prediction with respect to the predicted wind direction is shown in Fig 6 for both the split and 'fully' case of the CRTRL algorithm. It could be observed that wind blowing from different directions has different speed which results in different wind turbine power output. It is also worth noticing that while both methods capture the strongest wind directions, the split-based CRTRL algorithm predicts a slightly wider range of directions with strong wind. Similarly, the split-based power estimation mainly results in higher power estimates than a fully-complex one as shown in Fig 7.

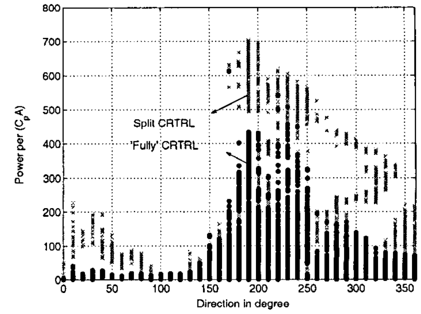

Fig. 6. Prediction of WT power with respect to wind direction

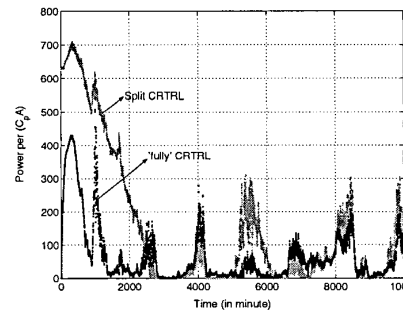

Fig. 7. Prediction of WT power with respect to time

V. Conclusion

The paper formulates and solves the problem of estimation of wind profile and wind turbine power in the complex domain taking into account the strong correlation of two wind components, wind speed and wind direction. A complex-valued pipelined recurrent neural network (CPRNN) for prediction of nonlinear and nonstationary signals has been used. The complex-valued real time recurrent learning (CRTRL) algorithm has been introduced for nonlinear adaptive filtering performed by RNNs in the complex domain, and has been derived for a general complex activation function of a neuron. Unlike the previous algorithms of this kind, the proposed CRTRL algorithm is generic and applicable for a variety of complex signals including those with strong component correlations. The performance of the CPRNN architecture has been evaluated on real-life wind signals and have shown to give high accuracy of wind profile and wind turbine power prediction.

References

[1] S.Li,D.C.Wunsch,E.A.O'Hair and M.Giesselmann, "Using NN to Estimate Wind Turbine Power Generation", *IEEE Trans on Energy Conversion,* Vol.16, No.3, Sep.2001, pp.276-282.

[2] M.C.Alexiadis et al, "Short-term Forecasting of Wind Speed and Related Electrical Power", *Solar Energy,* Vol.63, No.1, 1998, pp.61-68.

[3] Special Issue on *Neurocomputing,* Vol 23, No 1-3, 1998.

[4] D.Mandic and J.Chambers, *Recurrent Neural Networks for Prediction,* Wiley 2001.

[5] T. Kim and T. Adali, "Approximation By Fully Complex MLP Using Elementary Transcendental Activation Functions," in *In Proceedings of XI IEEE Workshop on Neural Networks for Signal Processing*, pp. 203-212, 2001.

[6] B. Widrow, J. McCool, and M. Ball, "The Complex LMS Algorithm," *Proceedings of the IEEE*, Vol. 63, 1975, pp. 712-720.

[7] R. J. Williams and D. A. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Computation*, Vol. 1, no. 2, 1989, pp. 270-280.

[8] S. Haykin and Liang Li, "Nonlinear adaptive prediction of nonstationary signals," *IEEE Transactions on Signal Processing*, Vol. 43, no. 2, 1995, pp.526-535.