The Relationship between Cyclostationarity, Sampling and Synchronization in Modern Communications Problems

PhD Candidacy Examination

Zikun Tan

Advisor: Prof. Ron Dabora

Department of Electrical & Computer Engineering
Ben-Gurion University of the Negev
Be'er Sheva, Israel

April 20, 2023

Outline

Outline

In many scenarios in *statistical signal processing* and *communications*, it is commonly assumed that the involved random processes have ***stationary statistics***.

However, man-made signals are typically generated using *periodic operations*, by which ***cyclostationarity*** is induced:

- The statistical characterization of cyclostationary processes depends on the *time* as well as the *lag*.
- Cyclostationarity is also widely observed in some other fields, e.g., geophysics, rotating machinery and econometrics, therefore it is a common statistical property.
- In many practical signal analysis problems, communications signals are modelled as ***wide-sense cyclostationary (WSCS) processes***.

WSCS processes:

Definition (WSCS processes [1] [2])

A continuous-time (CT) (resp. discrete-time (DT)) random process $X(t)$, $t \in \mathcal{R}$ (resp. $X[n]$, $n \in \mathcal{Z}$) is called ***WSCS*** if both its ***mean*** $m_X(t)$ (resp. $m_X[n]$) and its ***autocorrelation function*** $R_X(t, \tau)$ (resp. $R_X[n, \Delta]$) are periodic in time t (resp. n) with some period $T_0 \in \mathcal{R}$ (resp. $N_0 \in \mathcal{Z}$) for any lag value $\tau \in \mathcal{R}$ (resp. $\Delta \in \mathcal{Z}$).

[1] G. B. Giannakis, "Cyclostationary signal analysis," in *Digital Signal Processing Handbook*, V. K. Madisetti and D. B. Williams, Eds., Boca Raton, FL, USA: CRC Press, 1999, ch. 17

[2] W. A. Gardner, A. Napolitano, and L. Paura, "Cyclostationarity: Half a century of research," *Signal Processing*, vol. 86, no. 4, pp. 639–697, 2006

Analytical expressions of WSCS processes:

Under mild regularity conditions, the ***mean*** $m_X(t)$ (resp. $m_X[n]$) and the ***autocorrelation function (AF)*** $R_X(t,\tau)$ (resp. $R_X[n,\Delta]$) of the CT (resp. DT) WSCS process $X(t)$ (resp. $X[n]$) are expressed in the following:

$$m_X(t) \triangleq \mathbb{E}\{X(t)\} = m_X(t+T_0)$$
$$R_X(t,\tau) \triangleq \mathbb{E}\{X(t+\tau)X^*(t)\} = R_X(t+T_0,\tau)$$

$$m_X[n] \triangleq \mathbb{E}\{X[n]\} = m_X[n+N_0]$$
$$R_X[n,\Delta] \triangleq \mathbb{E}\{X[n+\Delta]X^*[n]\} = R_X[n+N_0,\Delta]$$

Cyclic quantities of CT WSCS processes:

The ***AF*** of a *CT WSCS process* $X(t)$ with the *cyclostationarity period* T_0 can be expanded into a Fourier series via:

$$R_X(t,\tau) = \sum_{k=-\infty}^{\infty} R_X^{\alpha_k}(\tau) e^{j2\pi\alpha_k t}$$

where $\alpha_k \triangleq \frac{k}{T_0}$, $k \in \mathcal{Z}$. The Fourier coefficients $R_X^{\alpha_k}(\tau)$ are referred to as the ***cyclic autocorrelation function (CAF)*** and α_k is referred to as the ***cyclic frequency***.

The *CAF* $R_X^{\alpha_k}(\tau)$ can be obtained via:

$$R_X^{\alpha_k}(\tau) = \frac{1}{T_0} \int_{t=-\frac{T_0}{2}}^{\frac{T_0}{2}} R_X(t,\tau) e^{-j2\pi\alpha_k t} dt$$

Cyclic quantities of DT WSCS processes:

For a *DT WSCS process* $X[n]$ with the *cyclostationarity period* N_0, the *AF* can be expanded into a Fourier series via:

$$R_X[n,\Delta] = \sum_{k=0}^{N_0-1} R_X^{\tilde{\alpha_k}}[\Delta] e^{j2\pi\tilde{\alpha_k}n}$$

where $\tilde{\alpha_k} \triangleq \frac{k}{N_0}$, $k \in \{0,1,\ldots,N_0-1\}$. The Fourier coefficients $R_X^{\tilde{\alpha_k}}[\Delta]$ are the ***CAF*** and $\tilde{\alpha_k}$ is referred to as the ***cyclic frequency***.

The *CAF* $R_X^{\tilde{\alpha_k}}[\Delta]$ can be computed via:

$$R_X^{\tilde{\alpha_k}}[\Delta] = \frac{1}{N_0} \sum_{n=0}^{N_0-1} R_X[n,\Delta] e^{-j2\pi\tilde{\alpha_k}n}$$

Almost periodic functions:

Definition (Almost periodic functions [3] [4])

A CT (resp. DT) function $X(t)$, $t \in \mathcal{R}$ (resp. $X[n]$, $n \in \mathcal{Z}$) is called ***almost periodic***, if for any $\epsilon > 0$, there exists an associated number $l_\epsilon \in \mathcal{R}^{++}$ (resp. $\tilde{l}_\epsilon \in \mathcal{N}$) which satisfies that for any $\alpha \in \mathcal{R}$ (resp. $\beta \in \mathcal{Z}$), there exists $\delta \in [\alpha, \alpha + l_\epsilon]$ (resp. $\Delta \in [\beta, \beta + \tilde{l}_\epsilon]$), such that $|X(t+\delta) - X(t)| \leqslant \epsilon$ (resp. $|X[n+\Delta] - X[n]| \leqslant \epsilon$).

[3] E. Abakasanga, N. Shlezinger, and R. Dabora, "On the rate-distortion function of sampled cyclostationary Gaussian processes," *Entropy*, vol. 22, no. 3, 2020

[4] F. Chérif, "A various types of almost periodic functions on Banach spaces: Part I," *International Mathematical Forum*, vol. 6, no. 19, pp. 921–952, 2011

An illustrative example [5] *of an almost periodic function*:

Consider function $x_1(t) = cos(\frac{2\pi t}{20})$ with period $T_1 = 20$, and function $x_2(t) = \frac{1}{4}cos(\frac{2\pi t}{\sqrt{2}})$ with period $T_2 = \sqrt{2}$.

- The ratio between T_1 and T_2, $\frac{20}{\sqrt{2}}$, is an *irrational number*.
- Therefore, function $x(t) = x_1(t) + x_2(t) = cos(\frac{2\pi t}{20}) + \frac{1}{4}cos(\frac{2\pi t}{\sqrt{2}})$ is ***not*** a periodic function. However, it is an ***almost periodic function***.

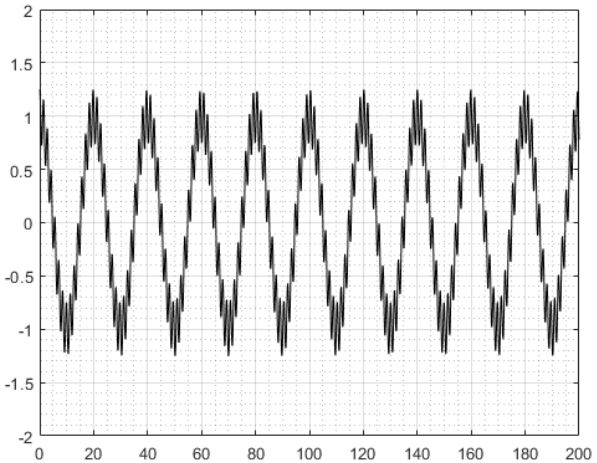

Figure 1: Plot of function $x(t) = cos(\frac{2\pi t}{20}) + \frac{1}{4}cos(\frac{2\pi t}{\sqrt{2}})$

[5] I. Amidror, *The Theory of the Moiré Phenomenon: Volume I: Periodic Layers*. Berlin, Germany: Springer, 2009

If we admit *a maximal* error $\epsilon = 0.3$, then it is given that $l_\epsilon = 20$ satisfies the definition of almost periodicity, namely, within a duration of 20, the function $x(t)$ repeats within 0.3 of its value at any selected time point:

- Solving $\text{argmin}_{\delta:0<\delta\leqslant 20} |X(t+\delta) - X(t)|$ by computer simulations, with the *resolution of* δ *0.1*, we obtain that $\delta^* = 19.8$.
  - $\delta^* = 19.8$ fits the definition of almost periodicity. However, with finer resolution of δ, a more accurate δ^* can be obtained.
- From the following plot, we see that the magnitude value of $\bar{x}(t) = x(t+19.8) - x(t)$ is always smaller than 0.3:

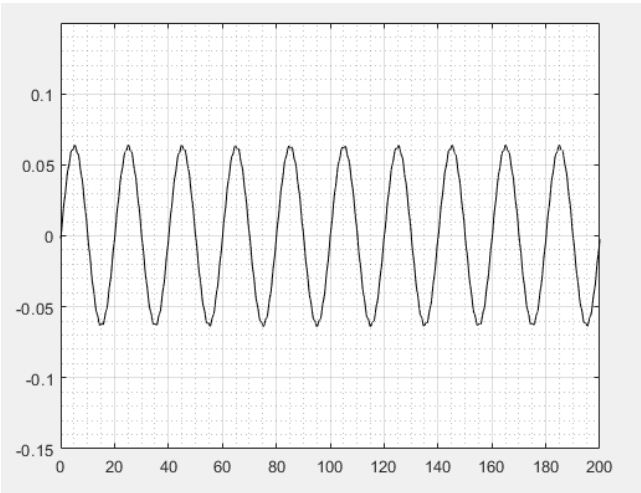

Figure 2: Plot of function $\bar{x}(t) = x(t+19.8) - x(t)$

- Here the value of $\alpha = 0$, but the ***almost periodicity*** holds for any $\alpha \in \mathcal{R}$.

If we admit a *smaller maximal error* $\epsilon = 0.01$, then it is given that $l_\epsilon = 140$ satisfies the definition of almost periodicity, namely, within a duration of 140, the function $x(t)$ repeats within 0.01 of its value at any selected time point:

- Solving $\text{argmin}_{\delta:0<\delta\leqslant 140} |X(t+\delta) - X(t)|$ by computer simulations, with the *resolution of* δ *0.1*, we obtain that $\delta^* = 140$.
  - $\delta^* = 140$ fits the definition of almost periodicity. However, with finer resolution of δ, a more accurate δ^* can be obtained.
- From the following plot, we see that the magnitude value of $\bar{x}(t) = x(t+140) - x(t)$ is always smaller than 0.01:

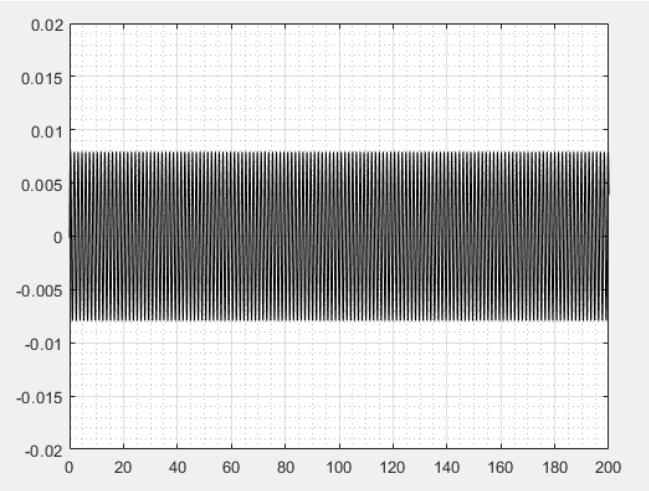

Figure 3: Plot of function $\bar{x}(t) = x(t+140) - x(t)$

- Here the value of $\alpha = 0$, but the ***almost periodicity*** holds for any $\alpha \in \mathcal{R}$.

Wide-sense almost cyclostationary (WSACS) processes:

Definition (WSACS processes [6] [7])

A CT (resp. DT) random process $X(t)$, $t \in \mathcal{R}$ (resp. $X[n]$, $n \in \mathcal{Z}$) is called ***WSACS*** if both its mean $m_X(t)$ (resp. $m_X[n]$) and its AF $R_X(t, \tau)$ (resp. $R_X[n, \Delta]$) are ***almost periodic*** in time t (resp. n) for any lag value $\tau \in \mathcal{R}$ (resp. $\Delta \in \mathcal{Z}$).

[6] G. B. Giannakis, "Cyclostationary signal analysis," in *Digital Signal Processing Handbook*, V. K. Madisetti and D. B. Williams, Eds., Boca Raton, FL, USA: CRC Press, 1999, ch. 17

[7] W. A. Gardner, A. Napolitano, and L. Paura, "Cyclostationarity: Half a century of research," *Signal Processing*, vol. 86, no. 4, pp. 639–697, 2006

Cyclic quantities of CT WSACS processes:

The *AF* of a *CT WSACS process* $X(t)$, $R_X(t,\tau)$, can be expanded into a generalized Fourier series:

$$R_X(t,\tau) = \sum_{\alpha\in\mathcal{A}} R_X^{\alpha}(\tau)e^{j2\pi\alpha t}$$

The Fourier coefficients $R_X^{\alpha}(\tau)$ are refered to as the ***CAF*** and $\mathcal{A}$ is a countable set of ***cyclic frequencies*** which are possibly ***incommensurate***.

The *CAF* $R_X^{\alpha}(\tau)$ can be obtained via:

$$R_X^{\alpha}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{t=-\frac{T}{2}}^{\frac{T}{2}} R_X(t,\tau)e^{-j2\pi\alpha t}dt$$

Cyclic quantities of DT WSACS processes:

For a DT WSACS process $X[n]$, its *AF* $R_X[n, \Delta]$ can be expanded into a generalized Fourier series as:

$$R_X[n, \Delta] = \sum_{\tilde{\alpha} \in \tilde{\mathcal{A}}} R_X^{\tilde{\alpha}}[\Delta] e^{j2\pi\tilde{\alpha}n}$$

$\tilde{\mathcal{A}}$ is a countable set of ***cyclic frequencies*** which are possibly ***incommensurate*** and the Fourier coefficients $R_X^{\tilde{\alpha}}[\Delta]$ are refered to as the ***CAF***.

The *CAF* $R_X^{\tilde{\alpha}}[\Delta]$ can be computed via:

$$R_X^{\tilde{\alpha}}[\Delta] = \lim_{N \to \infty} \frac{1}{2N+1} \sum_{n=-N}^{N} R_X[n, \Delta] e^{-2j\pi\tilde{\alpha}n}$$

Synchronous sampling & Asynchronous sampling:

Sampling a *CT WSCS process* does ***not*** necessarily result in a *DT WSCS process*.

When sampling a CT WSCS process, if its *cyclostationarity period* T_0 and the *sampling interval* T_{samp} satisfy a ***rational*** ratio, i.e.,:

$$\frac{T_0}{T_{samp}} = \frac{U}{V}$$

where $U, V \in \mathcal{N}$, then the resulting DT process is DT WSCS. This is referred to in this work as ***synchronous sampling***.

However, if T_0 and T_{samp} are related by an ***irrational*** ratio, i.e.,:

$$\frac{T_0}{T_{samp}} = \frac{U}{V} + \varepsilon;\ \varepsilon \in \mathcal{P} \text{ and } \varepsilon \in (0, 1)$$

then the resulting DT process is DT WSACS. This is referred to in this work as ***asynchronous sampling***.

Consider three cases of sampling a WSCS process $X(t)$, $t \in \mathcal{R}$ with variance $\sigma_{ct}^2(t)$:

- $T_{sym} = 1$ denotes the ***period*** of the CT variance, T_s denotes the ***sampling interval*** and ϕ denotes the ***sampling time offset (STO)***.
- The sampled DT process is $X[i] = X(i * T_{samp} + \phi)$.

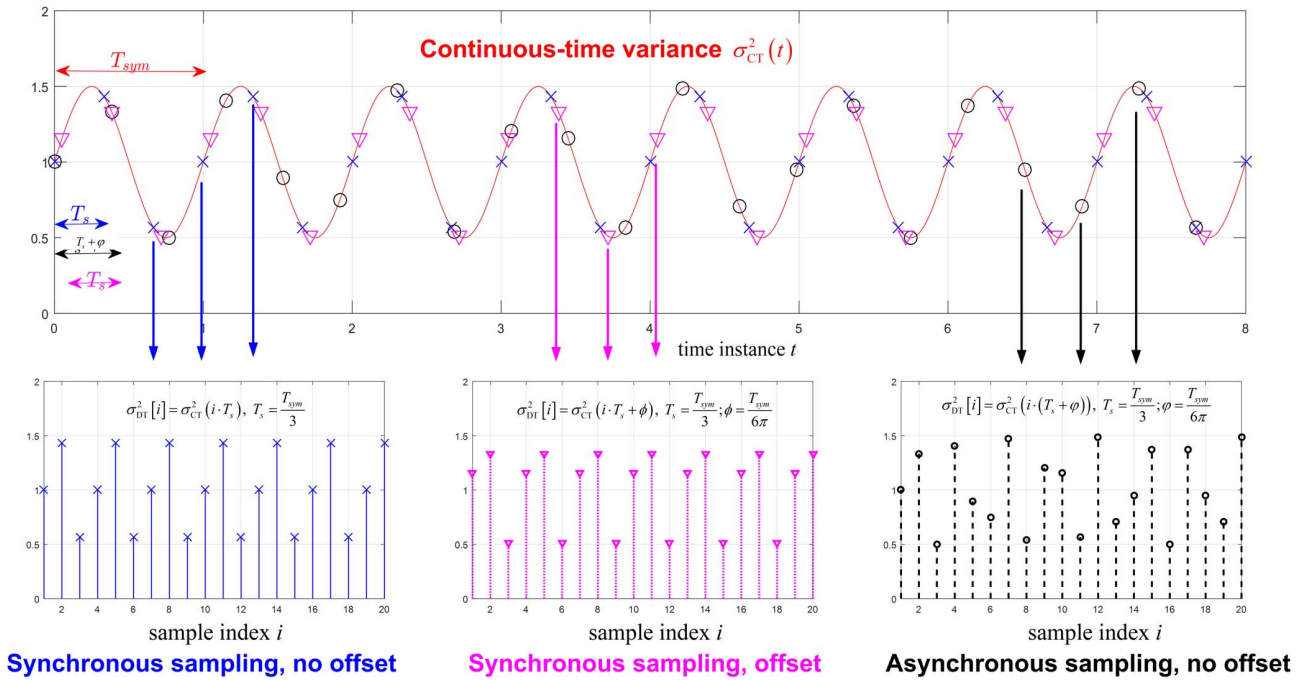

Figure 4: Three cases of sampling a WSCS process [8]

[8] N. Shlezinger *et al.*, "The capacity of memoryless channels with sampled cyclostationary Gaussian noise," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 106–121, 2020

- Applying sampling interval $T_s = \frac{T_{sym}}{3}$ and STO $\phi = 0$.
- The variance of the sampled DT process is periodic with a period of 3.
- The sampled process is a DT WSCS process. This represents ***synchronous sampling***.

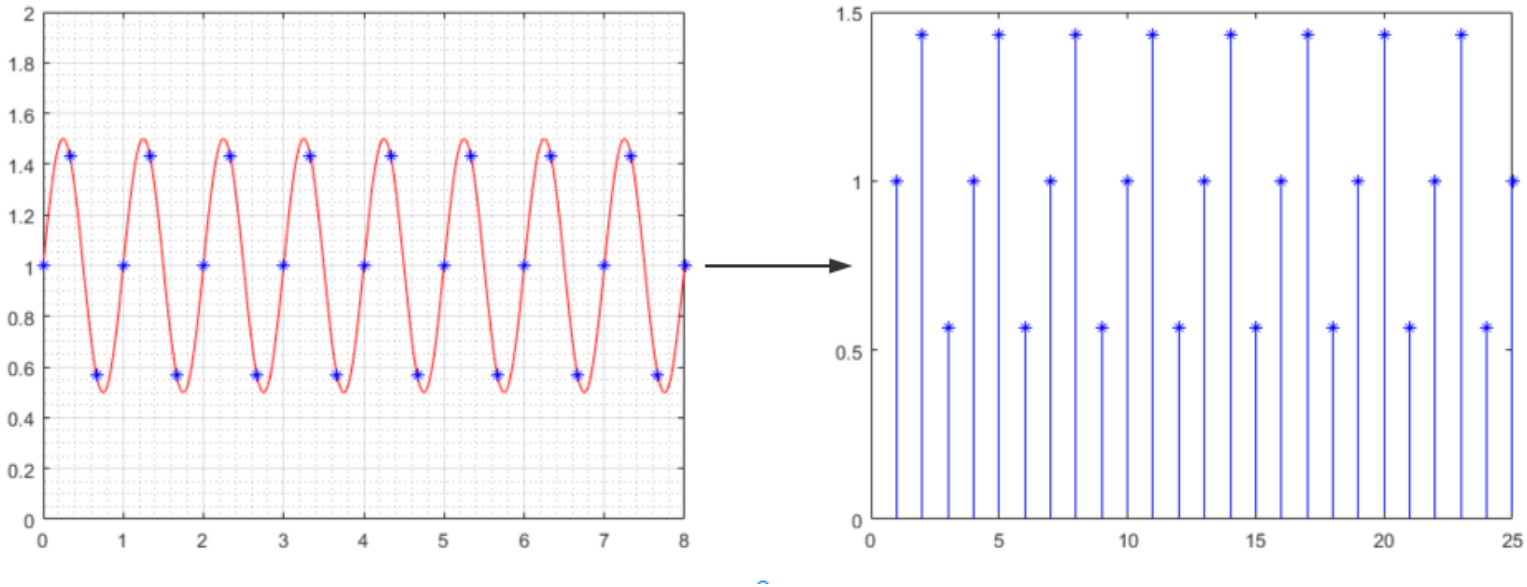

Figure 5: Case 1 for sampling a CT WSCS process

- Applying sampling interval $T_s = \frac{T_{sym}}{3}$ and STO $\phi = \frac{T_s}{2\pi}$.
- The variance of the sampled DT process is periodic with a period of 3.
- The sampled process is a DT WSCS process. This represents ***synchronous sampling***.

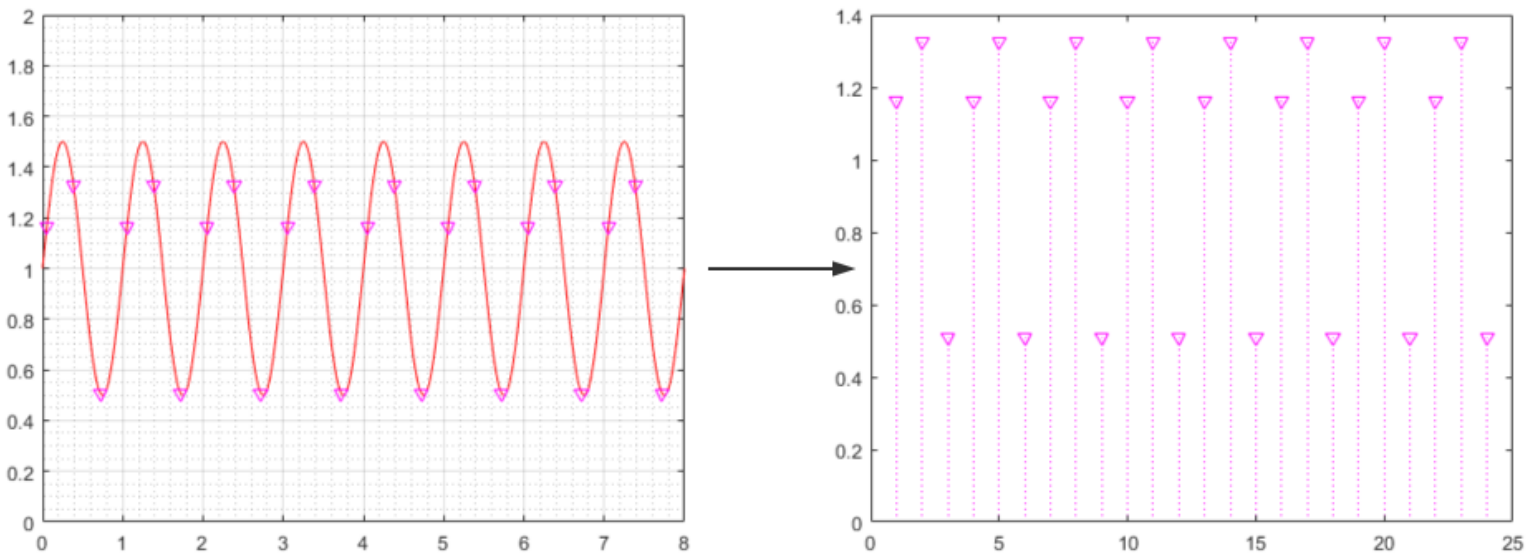

Figure 6: Case 2 for sampling a CT WSCS process

- Applying sampling interval $T_s = (1 + \frac{1}{2\pi})\frac{T_{sym}}{3}$ (ratio $\frac{T_{sym}}{T_s}$ is an irrational number) and STO $\phi = 0$.
- The variance of the sampled DT process is ***not*** periodic, but ***almost periodic***.
- The sampled process is a DT WSACS process. This represents ***asynchronous sampling***.

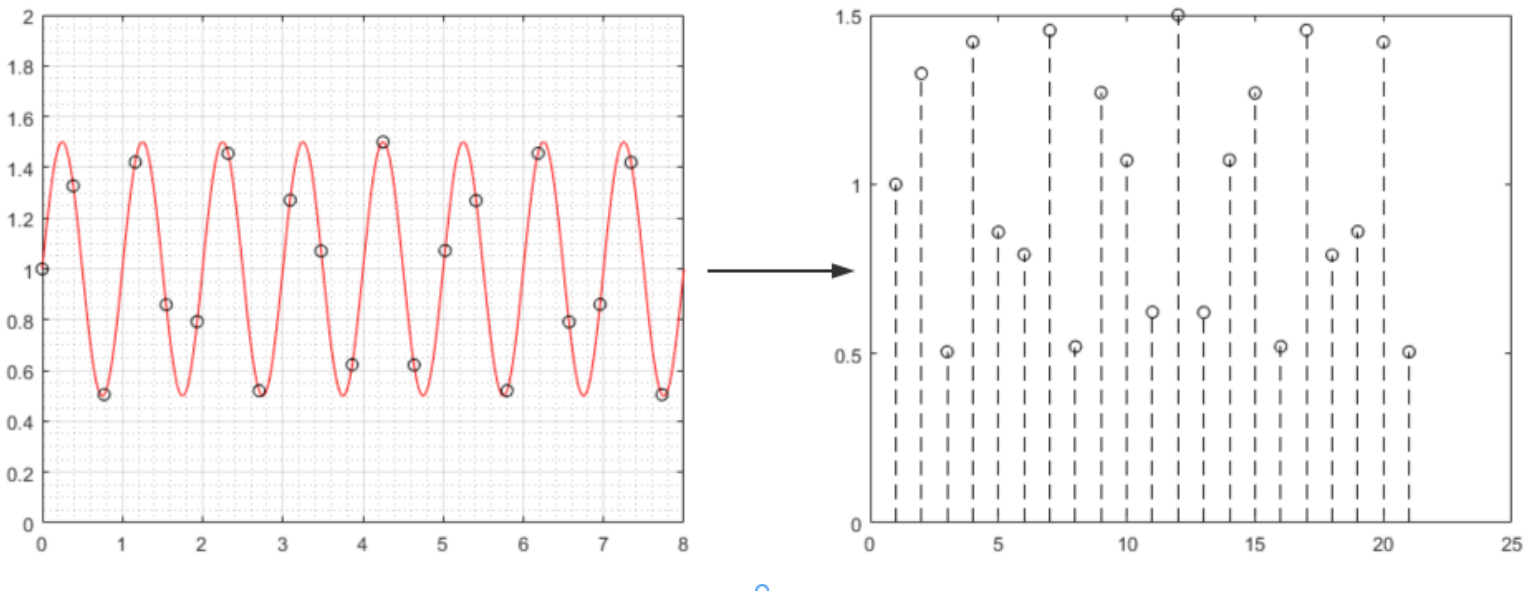

Figure 7: Case 3 for sampling a CT WSCS process

Outline

Deep learning is a subtype of ***machine learning***, which uses *neural networks* to learn the underlying mapping between the input and the output of a signal processing system.

Fully-connected neural networks (FCNNs), the most fundamental type of neural networks, where *neurons in each layer* are *fully connected* with *neurons in the next layer*:

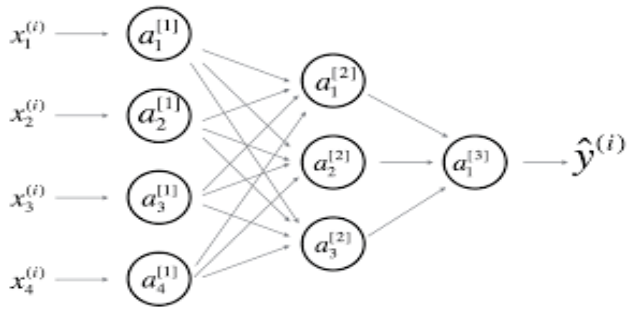

Figure 8: An FCNN with one input layer, two hidden layers and an output layer [9]

The relationship between quantities in each layer can be vectorially expressed via:

$$\mathrm{A}^{[1]} = g^{[1]}(\mathrm{W}^{[1]}\mathrm{X} + \mathbf{b}^{[1]})$$
$$\mathrm{A}^{[2]} = g^{[2]}(\mathrm{W}^{[2]}\mathrm{A}^{[1]} + \mathbf{b}^{[2]})$$
$$\hat{\mathrm{Y}} = \mathrm{A}^{[3]} = g^{[3]}(\mathrm{W}^{[3]}\mathrm{A}^{[2]} + \mathbf{b}^{[3]})$$

[9] A. Ng, *"CS230 Deep Learning: Standard notations for deep learning"*, Available online: https://cs230.stanford.edu/files/Notation.pdf, (Accessed on 2022-08-03)

- $\mathrm{X} = \begin{bmatrix} \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \cdots & \mathbf{x}^{(i)} & \cdots & \mathbf{x}^{(m)} \end{bmatrix}$ is the *input matrix*, where the superscripts (i) and (m) denote the *index of the input vector example* and the *total number of input vector examples*, respectively.
- $\mathrm{W}^{[l]}$ is the *weight matrix* of the l^{th} layer and there are three layers (i.e., only the hidden layers and the output layer are counted) in this FCNN.
- $\mathbf{b}^{[l]}$ denotes the *bias vector* of the l^{th} layer.
- $g^{[l]}(\cdot)$ denotes the *nonlinear activation function* (e.g., sigmoid function, tanh function, ReLu function, etc.) at the l^{th} layer and it is typically selected according to the application scenario.
- $\mathrm{A}^{[l]} = \begin{bmatrix} \mathbf{a}^{[l](1)} & \mathbf{a}^{[l](2)} & \cdots & \mathbf{a}^{[l](i)} & \cdots & \mathbf{a}^{[l](m)} \end{bmatrix}$ denotes the *output matrix* of the l^{th} layer.
- In this FCNN, the 3^{rd} layer is the *output layer*, therefore $\mathrm{A}^{[3]} = \hat{\mathrm{Y}} = \begin{bmatrix} \hat{\mathbf{y}}^{(1)} & \hat{\mathbf{y}}^{(2)} & \cdots & \hat{\mathbf{y}}^{(i)} & \cdots & \hat{\mathbf{y}}^{(m)} \end{bmatrix}$.

Training a neural network:

To train a neural network in the *supervised learning* setting:

- The ***loss function*** $L(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)})$ for the *i^{th} training example* represents the error between the *desired output label vector* $\mathbf{y}^{(i)}$ and the *actual output vector* $\hat{\mathbf{y}}^{(i)}$.
- The ***cost function*** $J(\mathrm{W}, \mathbf{b}) = \frac{1}{m} \sum_{i=1}^{m} L(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)})$ of the *entire training set* for the FCNN setting of W and $\mathbf{b}$, where W and $\mathbf{b}$ denote the concatenation of all $\mathrm{W}^{[l]}$ and the concatenation of all $\mathbf{b}^{[l]}$, respectively.
- In the ***backpropagation*** algorithm, the ***gradients*** of the ***cost function*** w.r.t. the weights and biases for each network layer are computed using the chain rule, then the weights and biases are updated using ***gradient descent*** to minimize the cost function.
- ***Adam*** algorithm, which combines the benefits of both ***adaptive gradient (AdaGrad)*** and ***root mean square propagation (RMSProp)***, is widely used in engineering applications to facilitate neural network training.

Convolutional neural networks (CNNs), which consist of multiple *convolutional layers*, *pooling (i.e., subsampling) layers* and *fully-connected layers*, are commonly used in processing *tensor signals*, e.g., image & video processing.

There are some special-purpose CNNs in engineering applications: LeNet-5, AlexNet, VGG-16 and residual networks (ResNets).

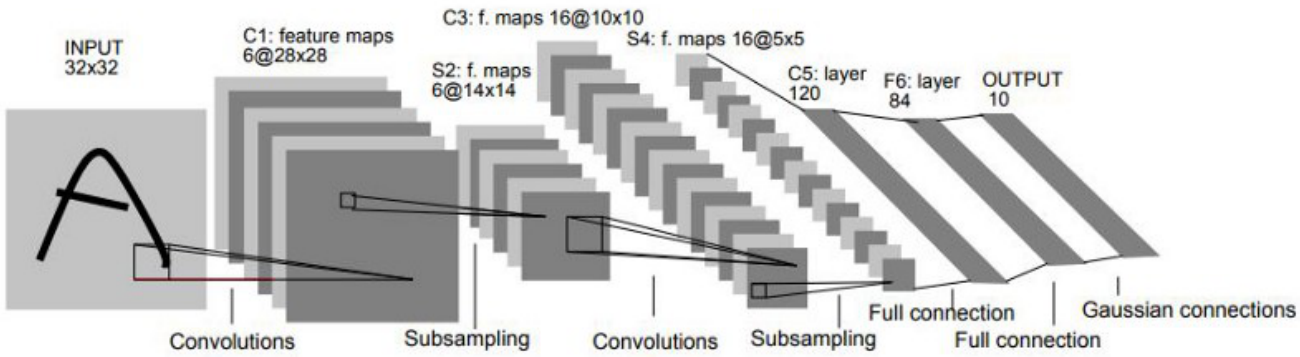

Figure 9: LeNet-5 for digit recognition [10]

[10] Y. LeCun *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998

Recurrent neural networks (RNNs) are widely used in signal processing problems involving *sequence models*, e.g., audio, speech & language processing.

An RNN consists of multiple RNN units concatenated in a *temporal sequence* with *hidden states* passed between each pair of consecutive units.

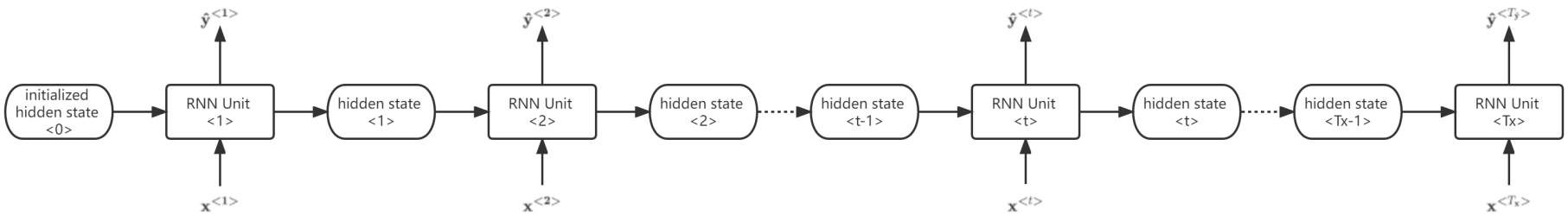

Figure 10: A unidirectional RNN

There are several possible types for the internal structure for an RNN unit: the ***Elman network*** unit, the ***Jordan network*** unit, the ***gated recurrent unit (GRU)*** unit and the ***long short-term memory (LSTM)*** unit

For an ***Elman network***, at timestep t, the relationship between the *input vector* $\mathbf{x}^{\langle t \rangle}$ and the *output vector* $\hat{\mathbf{y}}^{\langle t \rangle}$ of the RNN unit is stated in the following:

$$\mathbf{a}^{\langle t \rangle} = g_1(\mathrm{W}_a[\mathbf{a}^{\langle t-1 \rangle}, \mathbf{x}^{\langle t \rangle}] + \mathbf{b}_a) \tag{3a}$$

$$\hat{\mathbf{y}}^{\langle t \rangle} = g_2(\mathrm{W}_y \mathbf{a}^{\langle t \rangle} + \mathbf{b}_y) \tag{3b}$$

- The input activation vector $\mathbf{a}^{\langle t-1 \rangle}$ and the output activation vector $\mathbf{a}^{\langle t \rangle}$ are the *input hidden state* and the *output hidden state*, respectively.
- W_a and W_y are *internal weight matrices* while $\mathbf{b}_a$ and $\mathbf{b}_y$ are *internal bias vectors*.
- The *activation function* $g_1(\cdot)$ is usually a tanh function while the choice of the *output activation function* type of $g_2(\cdot)$ depends on the application scenario.

Jordan networks are similar to Elman networks and both of them are refered to as ***simple recurrent neural networks (SRNNs)***.

Typically, SRNNs do ***not*** properly capture *long-term dependencies* and suffer from the *vanishing gradient problem*, therefore ***GRU-based RNNs*** and ***LSTM-based RNNs*** are usually implemented in engineering applications.

If the RNN is based on ***GRUs***, then at the timestep t, the relationship between the *input vector* $\mathbf{x}^{\langle t \rangle}$ and the *output vector* $\hat{\mathbf{y}}^{\langle t \rangle}$ is presented in the following figure:

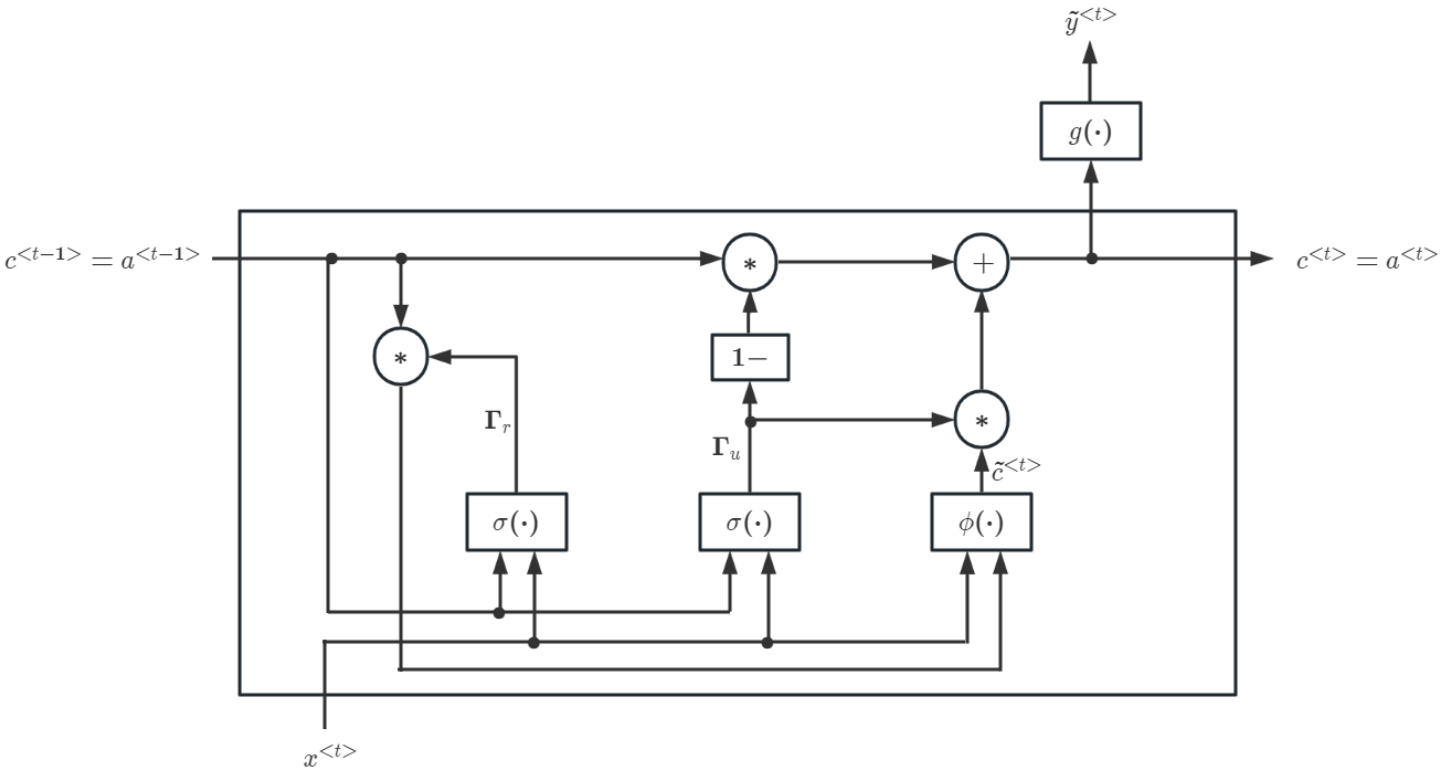

Figure 11: Internal structure of an GRU unit

Similarly, if the RNN is based on ***LSTM***, the relationship between the *input vector* $\mathbf{x}^{\langle t \rangle}$ and the *output vector* $\hat{\mathbf{y}}^{\langle t \rangle}$ is represented in the following figure:

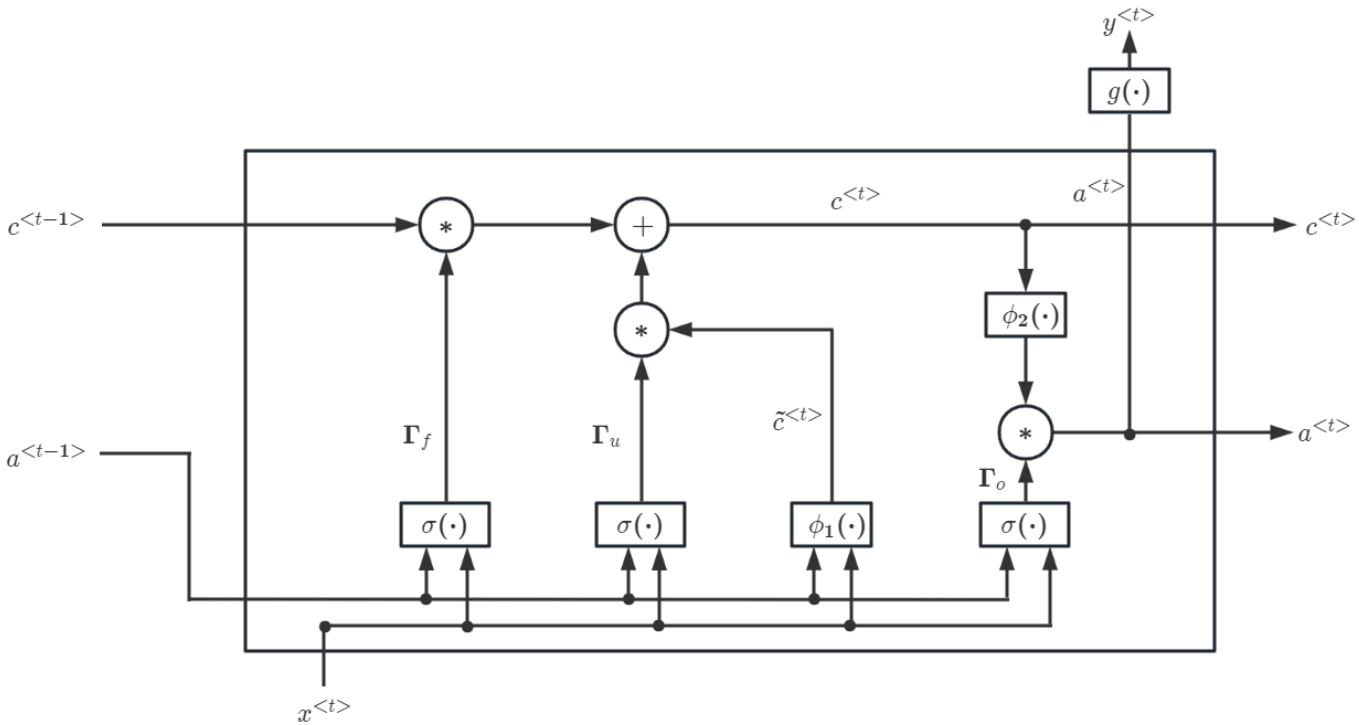

Figure 12: Internal structure of an LSTM unit

- $\mathbf{\Gamma}_u$, $\mathbf{\Gamma}_r$, $\mathbf{\Gamma}_f$ and $\mathbf{\Gamma}_o$ are refered to as the *update gate vector*, the *reset gate vector*, the *forget gate vector* and the *output gate vector*, respectively.
- $\tilde{\mathbf{c}}^{\langle t \rangle}$ denotes the *candidate memory cell*, $\mathbf{c}^{\langle t-1 \rangle}$ denotes the *input memory cell* and $\mathbf{c}^{\langle t \rangle}$ denotes the *output memory cell*.
- $\mathbf{a}^{\langle t-1 \rangle}$ denotes the *input activation vector* while $\mathbf{a}^{\langle t \rangle}$ denotes the *output activation vector*.
- $\sigma(\cdot)$ typically denotes the sigmoid function, $\phi(\cdot)$, $\phi_1(\cdot)$ and $\phi_2(\cdot)$ are usually tanh functions, where the choice of the output activation function $g(\cdot)$ depends on the application scenario.
- In both types of RNNs, $\mathbf{c}^{\langle t \rangle}$ is the *cell state* and $\mathbf{a}^{\langle t \rangle}$ is the *hidden state*. The cell state and the hidden state are ***always*** equal in *GRUs*-based RNNs but they are ***not*** necessarily equal in *LSTM*-based RNNs.

There is ***no*** straightforward rule about the choice of GRUs and LSTM in many engineering application problems.

However, generally, *GRU*-based RNNs are much ***simpler*** to implement and need ***shorter*** time and ***less*** computational power to train while *LSTM*-based RNNs are much more ***powerful*** in capturing long-term dependencies.

Unidirectional RNNs can also be extended to be ***bidirectional recurrent neural networks (BRNNs)*** and ***deep recurrent neural networks (DRNNs)*** to solve more sophisticated sequence model problems.

Outline

Definition (Source coding scheme [11] [12])

A ***source coding scheme*** $(2^{nR}, n)$ with the ***code rate*** R and the ***blocklength*** n consists of:

- An ***encoding function*** (also refered to as ***encoder***) $f_n(\cdot)$ that maps a sequence of n source symbols $\{x_i\}_{i=1}^n \equiv x^n$, over corresponding alphabets $\{\mathcal{X}_i\}_{i=1}^n \equiv \mathcal{X}^n$, into an index selected from a set containing 2^{nR} indices, i.e., $f_n(\cdot) : \mathcal{X}^n \to \{1, 2, \ldots, 2^{nR}\}$.
- A ***decoding function*** (also refered to as ***decoder***) $g_n(\cdot)$ that assigns a sequence of n reconstruction symbols $\{\hat{x}_i\}_{i=1}^n \equiv \hat{x}^n$, over corresponding alphabets $\{\hat{\mathcal{X}}_i\}_{i=1}^n \equiv \hat{\mathcal{X}}^n$, to each received index, i.e., $g_n(\cdot) : \{1, 2, \ldots, 2^{nR}\} \to \hat{\mathcal{X}}^n$.

- The set of codewords $\{g_n(i)\}_{i=1}^{2^{nR}}$ is referred to as the ***codebook***.
- $\{f_n^{-1}(i)\}_{i=1}^{2^{nR}}$ are the corresponding ***assignment regions***.

[11] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: John Wiley & Sons, 2006

[12] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, UK: Cambridge University Press, 2011

In *lossless source coding*, the ***probability of error*** of a $(2^{nR}, n)$ source code is defined as $P_e^{(n)} \triangleq \Pr\{\hat{X}^n \neq X^n\}$. If there exists a source code $(2^{nR}, n)$ which achieves $\lim_{n\to\infty} P_e^{(n)} = 0$, then:

- The code rate R is said to be ***achievable***.
- The ***optimal code rate*** R^* is the infimum of all achievable code rates

Theorem (Lossless source coding theorem for discrete memoryless sources (DMSs) [13])

A DMS X with entropy $H(X)$ can be encoded with a code rate ***not*** less than $H(X)$ with a negligible probability of error as $n \to \infty$; conversely if the code rate is less than $H(X)$, having reconstruction errors has a finite positive probability.

[13] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge, UK: Cambridge University Press, 2003

Lossless source coding can be used to compress *digital sources* whose symbols are drawn from *finite alphabets*.

However, lossless compression of *digital sources* whose symbols are drawn from *infinite alphabets* and of *analog sources* requires an infinite code rate, which is practically impossible. Therefore, for such sources, compression is applied with a bounded reconstruction error:

- This reconstruction error is characterized by a ***distortion measure function***, which, for a *single letter*, maps a *pair of source symbol alphabet* and *reconstruction symbol alphabet* into the *set of nonnegative real numbers*, i.e.,

$$d(\cdot,\cdot) : \mathcal{X} \times \hat{\mathcal{X}} \to \mathcal{R}^+$$

- The ***Hamming distortion function*** and the ***squared-error distortion function*** are two common distortion measure functions used to quantify the cost of representing the source symbol x by the reconstruction symbol $\hat{x}$:
  - Hamming distortion function:

$$d(x,\hat{x}) = \begin{cases} 0 & \text{if } x = \hat{x} \\ 1 & \text{if } x \neq \hat{x} \end{cases}$$

  - Squared-error distortion function:

$$d(x,\hat{x}) = (x-\hat{x})^2$$

The definition of distortion measure functions can be extended to *sequences*, by considering pairwise average distortions between the elements in a *sequence of source symbols* x^n and the elements in the corresponding *sequence of reconstruction symbols* $\hat{x}^n$:

$$d(x^n, \hat{x}^n) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i)$$

The ***expected distortion*** associated with a *lossy source coding scheme* $(2^{nR}, n)$ is defined as:

$$\mathbb{E}\{d(X^n, \hat{X}^n)\} = \mathbb{E}\left\{ d\left(X^n, g_n\left(f_n(X^n)\right)\right)\right\} = \sum_{x^n \in \mathcal{X}^n} p_{X^n}(x^n) d\left(x^n, g_n\left(f_n(x^n)\right)\right)$$

A ***rate-distortion pair*** (R, D) is said to be ***achievable*** if there exists a sequence of source codes $(2^{nR}, n)$ which satisfies $\lim_{n\to\infty} \mathbb{E}\left\{ d\left(X^n, g_n\left(f_n(X^n) \right) \right) \right\} \leqslant D$

The ***rate-distortion region*** is the closure of the set of achievable rate-distortion pairs

The ***rate-distortion function (RDF)***, denoted as $R(D)$, is the infimum of the code rates R for which the rate-distortion pair (R, D) is achievable

Definition (Information RDF [14])

For a DMS X defined over $\mathcal{X}$, a reconstruction random variable $\hat{X}$ defined over $\hat{\mathcal{X}}$, and a distortion measure function $d(x, \hat{x})$, the ***information RDF*** $R^{(I)}(D)$ is defined as:

$$R^{(I)}(D) = \min_{p_{\hat{X}|X}(\hat{x}|x):\mathbb{E}\{d(X,\hat{X})\}\leqslant D} I(X;\hat{X})$$

where $I(\cdot;\cdot)$ denotes the mutual information between two random variables and the minimization is over all conditional distributions $p_{\hat{X}|X}(\hat{x}|x)$ which satisfies the preset distortion constraint $\mathbb{E}\{d(X,\hat{X})\} \leqslant D$.

[14] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: John Wiley & Sons, 2006

For *DMSs*, the *RDF* is equal to the *information RDF*. Therefore the ***lossy source coding theorem*** can be expressed in the following:

Theorem (Lossy source coding theorem [15] [16])

For a DMS X defined over $\mathcal{X}$, a reconstruction random variable $\hat{X}$ defined over $\hat{\mathcal{X}}$, and a distortion measure function $d(x, \hat{x})$, the RDF is equal to the respective information RDF:

$$R(D) = R^{(I)}(D) = \min_{p_{\hat{X}|X}(\hat{x}|x):\mathbb{E}\{d(X,\hat{X})\}\leqslant D} I(X;\hat{X})$$

where $R(D)$ is the infimum of all code rates for which $\mathbb{E}\{d(X, \hat{X})\} \leqslant D$.

[15] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: John Wiley & Sons, 2006

[16] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, UK: Cambridge University Press, 2011

Two examples for the application of the lossy source coding theorem:

- The RDF for compressing a *DMS* $\sim$ *Bern*(p) with the *Hamming distortion function* is expressed as:

$$R(D) = \begin{cases} H_b(p) - H_b(D) & 0 \leqslant D \leqslant \min\{p, 1-p\} \\ 0 & D > \min\{p, 1-p\} \end{cases}$$

  where $H_b(\cdot)$ denotes the ***binary entropy function***.

- The RDF for compressing a *DMS* $\sim \mathcal{N}(0, \sigma^2)$ with the *squared-error distortion function* is expressed as:

$$R(D) = \begin{cases} \frac{1}{2} \log \frac{\sigma^2}{D} & 0 \leqslant D \leqslant \sigma^2 \\ 0 & D > \sigma^2 \end{cases}$$

More generally, when compressing m independent (but ***not*** necessarily identical) Gaussian sources $X_1, X_2, \ldots, X_m$, $X_i \sim \mathcal{N}(0, \sigma_i^2)$ and using the *squared-error distortion function*, the RDF is:

$$R(D) = \begin{cases} \frac{1}{2} \sum_{i=1}^{m} \log \frac{\sigma_i^2}{D_i} & 0 \leqslant D \leqslant \sum_{i=1}^{m} \sigma_i^2 \\ 0 & D > \sum_{i=1}^{m} \sigma_i^2 \end{cases}$$

where $D_i = \min\{\sigma_i^2, \lambda\}$ and λ is chosen for $\sum_{i=1}^{m} D_i = D$.

Outline

Automatic modulation classification (AMC) refers to the identification of the modulation scheme used for generating a communications signal via analysis of the received signal at the receiver.

- AMC is a step between *signal detection* and *demodulation*.
- AMC is widely used in modern communications systems and spectrum monitoring systems.

The design of an automatic modulation classifier generally involves ***two*** steps [17]:

1. ***Preprocessing*** (e.g., noise reduction, carrier frequency estimation and equalization).
2. The ***modulation classification rule*** applied based on the preprocessed information.
   - Preprocessing typically depends on the design of the modulation classification rule in the second step.

[17] O. A. Dobre, A. Abdi, and W. Su, "Survey of automatic modulation classification techniques: Classical approaches and new trends," *IET Communications*, vol. 1, pp. 137–156, May 2007

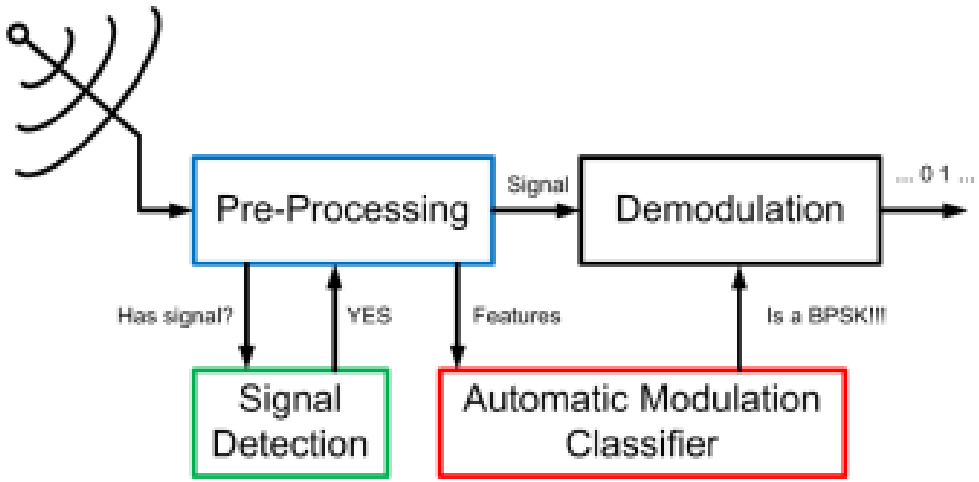

Figure 13: Structure of an automatic modulation classifier [18]

[18] A. I. R. Fontes *et al.*, "Automatic modulation classification using information theoretic similarity measures," in *2012 IEEE Vehicular Technology Conference (VTC Fall 2012)*, Québec City, QC, Canada, Sep. 2012, pp. 1–5

Generally, AMC algorithms can be classified into ***likelihood-based (LB)*** methods and ***feature-based (FB)*** methods [19]:

- ***LB*** methods are based on the *likelihood function* of the received signal and the final classification decision is reached by comparing the likelihood ratio with a *threshold level*.
- ***FB*** methods are based on extracting *useful characteristic features* from the received signal which are the basis for the classification decision.

A modulation classifier should:

- Decide which modulation scheme is used to generate the received signal,
- or conclude that the received modulation scheme is ***not*** within the predefined classification catalog.

[19] O. A. Dobre, A. Abdi, and W. Su, "Survey of automatic modulation classification techniques: Classical approaches and new trends," *IET Communications*, vol. 1, pp. 137–156, May 2007

An example work of AMC:

- [Câmara et al.: *IEEE Access* 2019] proposed an AMC architecture for a set of digital modulations contaminated by additive impulsive noise based on cyclostationary features.
  - The CAF, the fractional lower-order cyclic autocorrelation function (FLOCAF) and the cyclic correntropy function (CCF) were used to extract cyclostationary features.
  - The CAF feature did not facilitate the desired AMC performance, while the FLOCAF and CCF features did.
  - The CCF-based scheme always performed better than that of the FLOCAF-based scheme.

Outline

We have already shown that sampling a *CT WSCS process* does ***not*** necessarily result in a *DT WSCS process*:

- When the *sampling interval* is related with the *cyclostationarity period* via a ***rational*** ratio, the resulting process is a ***DT WSCS process***. This is referred to as ***synchronous sampling***.
- When the *sampling interval* is related with the *cyclostationarity period* via an ***irrational*** ratio, the resulting process is a ***DT WSACS process***. This is referred to as ***asynchronous sampling***.

When does the sampled DT process ***have memory*** and when it ***do not***?

- When the *sampling interval* is ***larger*** than the *maximal correlation length* of the CT process, the resulting DT process is ***memoryless***.
- Otherwise, the resulting DT process is ***with memory***.

Previous relevant works:

- [Kipnis et al.: *IEEE TIT* 2018] characterized the ***distortion-rate function (DRF)*** of both CT and DT WSCS Gaussian sources *with memory*.
- Based on [Kipnis et al.: *IEEE TIT* 2018], [Abakasanga et al.: *Entropy* 2020] characterized the *RDF* of DT WSACS *memoryless* Gaussian sources, using the ***information-spectrum framework*** [20].
  - *DT WSACS sources* are ***not*** information stable, thus the *information-spectrum framework* is employed instead of conventional information-theoretic tools.
  - It is assumed that the *target distortion constraint* is ***less*** than the *minimum variance* of the original CT WSCS Gaussian source.

Our proposed topic:

- ***Characterize the RDF of DT WSACS Gaussian sources with memory***.

[20] T. S. Han, *Information-Spectrum Methods in Information Theory*. Berlin, Germany: Springer, 2003

Motivation Example of Our Research:

Compress-and-forward (CF) relay networks:

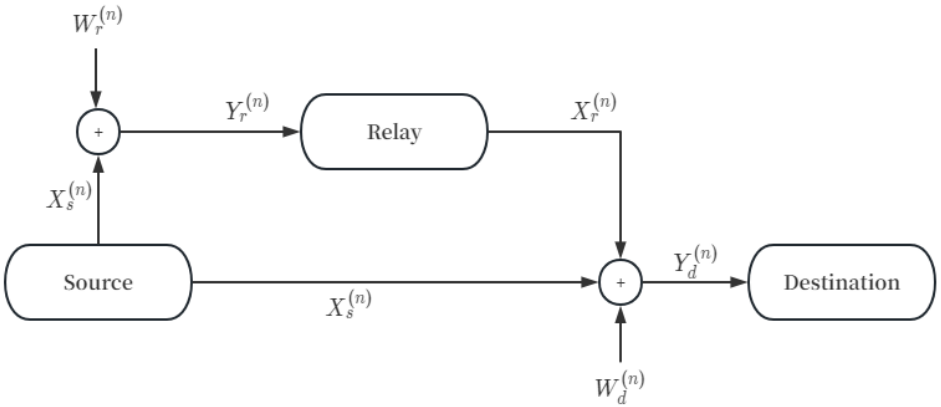

Figure 14: A CF relay network

- The signals received at the *relay* and at the *destination* are essentially different noisy versions of the ***same*** source signal, thus they are ***correlated***.
- The signal received at the *relay* is ***compressed*** and ***forwarded*** to the *destination*.
- The destination decodes the information from the ***joint*** output of the ***source-destination channel*** and the ***relay-destination channel***.

Applying *compression* at the *relay* requires the relay to first sample its received signal, which is practice is a communications signal, e.g., an orthogonal frequency-division multiplexing (OFDM) signal. Then:

- Practically, as the ***oscillators*** at the *relay* and at the *source* are ***physically separate***, then, due to the ***inherent variability*** of oscillators' frequencies, it is practically reasonable that, the *sampling interval* at the relay is related to the *cyclostationarity period* of the received communications signal via an ***irrational*** ratio.

- Moreover, the *sampling interval* at the relay is typically ***smaller*** than the *maximal correlation length* of the received communications signal.

Therefore, the sampled received communications signal at the relay is typically modelled as a ***DT WSACS Gaussian source with memory***.

Our Future Research Plan:

Research starting points:

- [Dabora and Abakasanga: *Submitted to IEEE TIT* 2022] characterized the *capacity* of asynchronously sampled linear channels modelled as additive CT WSCS Gaussian noise *with memory*, using the ***information-spectrum framework***.
  - This work generalized the model used in [Shlezinger et al.: *IEEE TCOM* 2020] to add ***memory*** to the asynchronously sampled WSCS Gaussian interference model.
- [Abakasanga et al.: *Entropy* 2020] characterized the *RDF* of DT WSACS ***memoryless*** Gaussian sources, using the ***information-spectrum framework***.

Our topic is a ***dual problem*** of the channel capacity characterization problem considered in [Dabora and Abakasanga: *Submitted to IEEE TIT* 2022].

Following the *general outline* of [Abakasanga et al.: *Entropy* 2020], we could adapt some of the tools and analysis methods from [Dabora and Abakasanga: *Submitted to IEEE TIT* 2022] for our research.

Outline

AMC tasks have been extensively explored using ***deep learning (DL)*** and ***cyclostationary properties***.

To improve on the performance of previous AMC approaches, in our research, we intent to address AMC tasks using the following three elements:

- Reinforcement learning (RL)
- Multi-antenna reception
- Eye patterns

Before introducing the basic principles of RL, recall the representation of the agent-environment interaction as in a Markov decision process (MDP):

1. At ***timestep*** t: The agent observes the environmental state (i.e., ***state*** S_t) and selects an action (i.e., ***action*** A_t) in response to the observed environment, based on the *current policy*.
2. At ***timestep*** $t+1$: The agent receives the ***reward*** R_{t+1} from the environment and the next ***state*** S_{t+1} is induced.

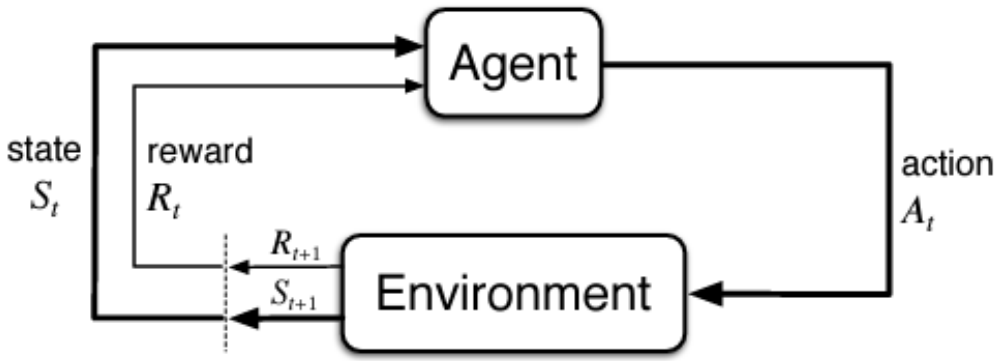

Figure 15: The agent–environment interaction in an MDP [21]

[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning - An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

The basic principles of ***RL***:

- RL is one of the three paradigms of *machine learning*, alongside ***supervised learning*** and ***unsupervised learning***.
- In RL, the agent learns to map ***environmental observations*** to different ***actions*** to maximize a *long-term* cumulative ***reward***.
- RL implements ***goal-directed learning*** from ***trial-and-error*** experiments with the environment.
- The RL optimization problem is typically modelled as an ***MDP***, or more commonly, a ***partially observable Markov decision process (POMDP)***.
- ***Dynamic programming*** techniques are usually adopted for problem-addressing in MDPs and POMDPs.
- RL is extensively applied in many *sequential decision-making problems* under *uncertainty* (e.g., robotics & industrial automation).

Previous relevant works:

- [Câmara et al.: *IEEE Access* 2019] (reviewed previously)
- [Dobre et al.: *Wireless Pers Commun* 2010] proposed an AMC method for distinguishing between linear digital modulations based on higher-order cyclic cumulants (CCs) in flat-fading channels.
  - This method can be implemented in both single-antenna and multi-antenna classifiers.
  - In the single-antenna CC-based classifier, the estimated feature vector containing the absolute values of several CCs of the sampled baseband signal received over a flat-fading channel is computed.
  - Each modulation scheme has its own analytical feature vector.
  - The modulation scheme whose analytical feature vector is closest with the estimated feature vector is decided to be the classification result.
  - In the multi-antenna CC-based classifier, a selection combiner is applied to select the received signal with the highest signal-to-noise ratio (SNR), then the selected received signal is used to make the modulation classification as in the single-antenna classifier.

Our proposed plan for solving AMC tasks using RL:

- We propose to begin with a binary modulation classification problem
  1. The agent computes the decision metric for the two modulations via a deep neural network (DNN)
  2. If the metric is higher than a threshold, the agent selects the corresponding modulation as the classification result; otherwise the classification result is the other modulation
  3. The received signal is demodulated using the decided modulation scheme
  4. Check the integrity of the demodulated information using a cyclic redundancy check (CRC)
  5. If the CRC test is passed, give the agent positive rewards; otherwise, give the agent negative rewards
- Then proceed to a multiclass modulation classification problem.
- Some other aspects for consideration, e.g., determining the reward without demodulating the received signal to protect information privacy.

Other two elements:

- Inspired by [Dobre et al.: *Wireless Pers Commun* 2009], by receive diversity, use a weighted combination of the output of each receiving antenna or a weighted combination of the classification evaluation metric from each receiving antenna.
- Implement AMC using eye patterns:
  - Minimal inter-symbol interference (ISI) and channel noise lead to the "opening" of an eye pattern while the "closure" of an eye pattern is the result of severe ISI and channel noise.
  - DNN could be applied to perform the classification task, possibly based on a processed version of the eye pattern of the received signal.

Outline

In multi-user communications, multiple-access techniques are applied to facilitate sharing the common and finite wireless radio resources

Conventionally, orthogonal multiple access (OMA) schemes are used:

- Frequency-division multiple access (FDMA)
- Time-division multiple access (TDMA)
- Code-division multiple access (CDMA)
- Space-division multiple access (SDMA)

From 1G to 4G communications, FDMA, TDMA, CDMA and orthogonal frequency-division multiple access (OFDMA) were applied, respectively

However, non-orthogonal multiple access (NOMA) has been a promising multiple-access technique for the ongoing 5G communications and beyond

The basic principles of a NOMA network:

- All users transmit their signals ***simultaneously over the same frequency band*** to a common destination receiver.
- The destination receiver receives the superposition of these transmitted signals.

Advantages of NOMA over OMA:

- Larger system capacity.
- High energy/spectral efficiency.
- Low latency.

Two fundamental algorithms for decoding the received signal for each user:

- Joint decoding:
  - Decode all received signals simultaneously.
  - This algorithm typically leads to good performance, but its implementation usually requires high computational complexity.
- Successive interference cancellation (SIC):
  1. Receive a mixed signal consisting of several signals with different power levels.
  2. Decode the stronger signal, while regarding the weaker signals as the noise.
  3. Reconstruct the stronger signal and subtract it from the received signal.
  4. If not all signals are decoded, go to Step 2.

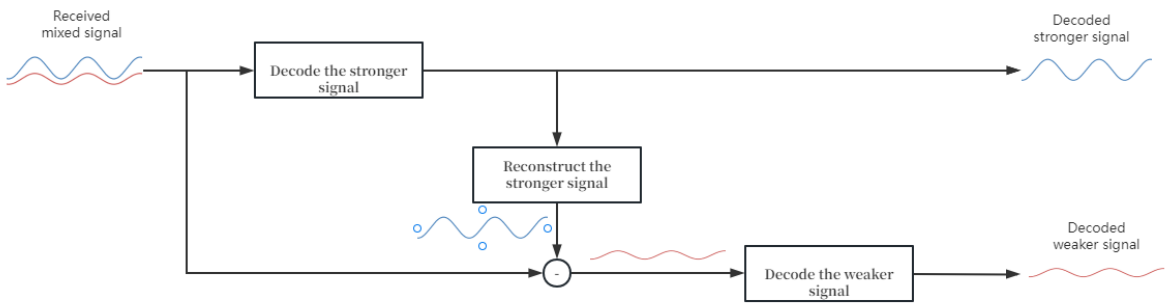

Figure 16: SIC

A motivating NOMA network for consideration:

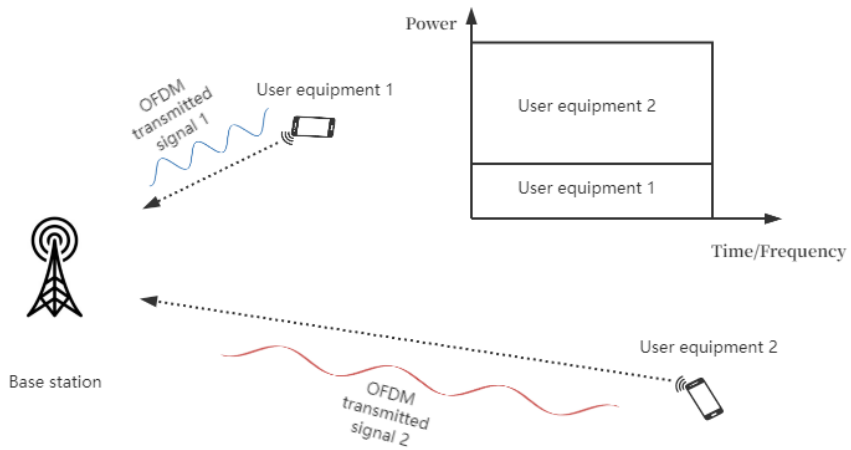

Figure 17: A NOMA network

- User equipment (UE) 1 and UE 2 transmit their OFDM signals simultaneously in the same frequency band towards a common base-station.
- Each OFDM signal is generated with its own sampling interval in the respective UE.
- To implement either decoding algorithms (i.e., joint decoding or SIC), the base-station needs to correctly sample each received OFDM signal, to avoid inter-carrier interference (ICI)/ISI.
- Therefore, we need to perform sampling frequency synchronization in the base-station for two signals generated with different clocks.

To the best of our knowledge, the problem of sampling frequency synchronization in NOMA networks has not been considered previously.

However, we could use works about sampling frequency synchronization in OMA networks as starting points:

- [Kumar and Dabora: *IEEE Access* 2019] implemented a blind sampling frequency offset (SFO) estimation based on cyclostationary properties in OFDM systems, which is refered to as the CB-SFS algorithm.
  - Using a finite-length sequence of received signal samples, a cost function corresponding to the sum of autocorrelated cyclic autocorrelation functions (ACAFs) over a range of time lags is empirically computed at every possible cyclic frequency.
  - The location of the local maxima of the ACAF corresponds to the cyclic frequency value of the sampled received signal, thus the fundamental cyclic frequency can be obtained.
  - Then, using the ratio relationship between the fundamental cyclic frequency and the sampling interval, the sampling interval at the transmitter is acquired.
  - With the known sampling interval at the receiver, the value of SFO is finally computed.

- [Jung et al.: *IEEE Access* 2018] proposed an unbiased least squares (LS) algorithm with low computational complexity for the joint estimation of the carrier frequency offset (CFO) and the SFO in an OFDM system.
  - Continual pilots are non-uniformly distributed pilot symbols embedded in the OFDM symbols in the DVB-C2 system.
  - The proposed algorithm computes the phase-difference dependent signals between a pair of successive OFDM symbols for a specifically selected set of continual pilots.
  - Then, two principal phase angles are obtained as two sums of these phase differences.
  - The estimates of the CFO and the SFO are finally computed from the two principal phase angles via some simple operations.

- [Rotem and Dabora: *IEEE Access* 2020] proposed a sequential and pilot-assisted algorithm for the joint estimation of the SFO and the CFO in OFDM systems.
  - For the SFO estimation, the algorithm first makes some processing of the pilot subcarriers from any two consecutive OFDM symbols to form a cost function containing only one auxiliary parameter. Then the SFO estimate is obtained by substituting two designated values of the auxiliary parameter into the cost function and applying some simple operations.
  - Then the CFO estimation, depending on either of the two proposed CFO estimator, is made based on the successful SFO compensation.


THANK
YOU!

Outline

Analytical CAF:

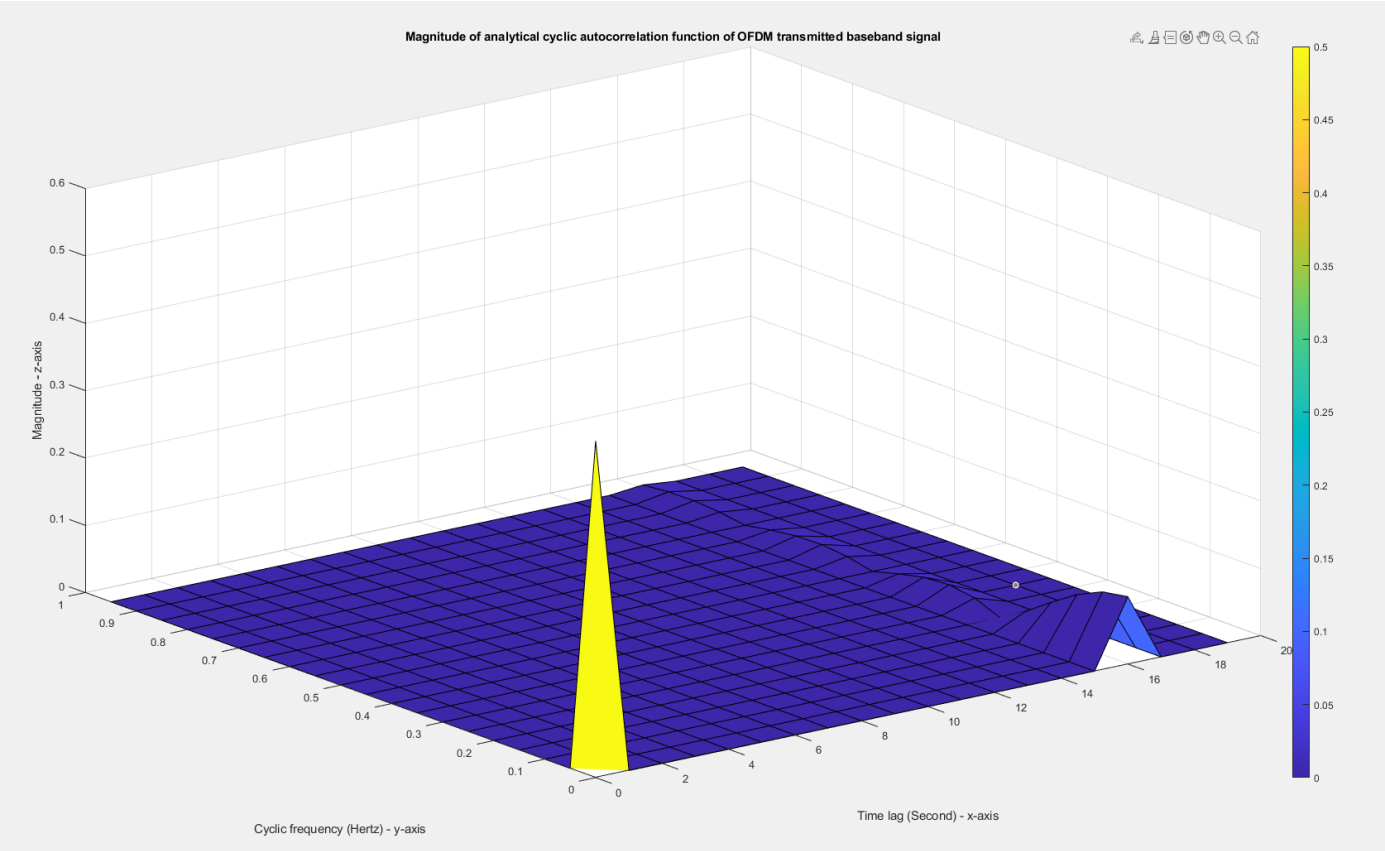

Figure 18: Magnitude of the analytical CAF of an OFDM signal with subcarriers modulated by BPSK (there are 5 OFDM symbols, each symbol has 16 subcarriers and 4 cyclic prefixes with the sampling interval 1s)

Empirical CAF:

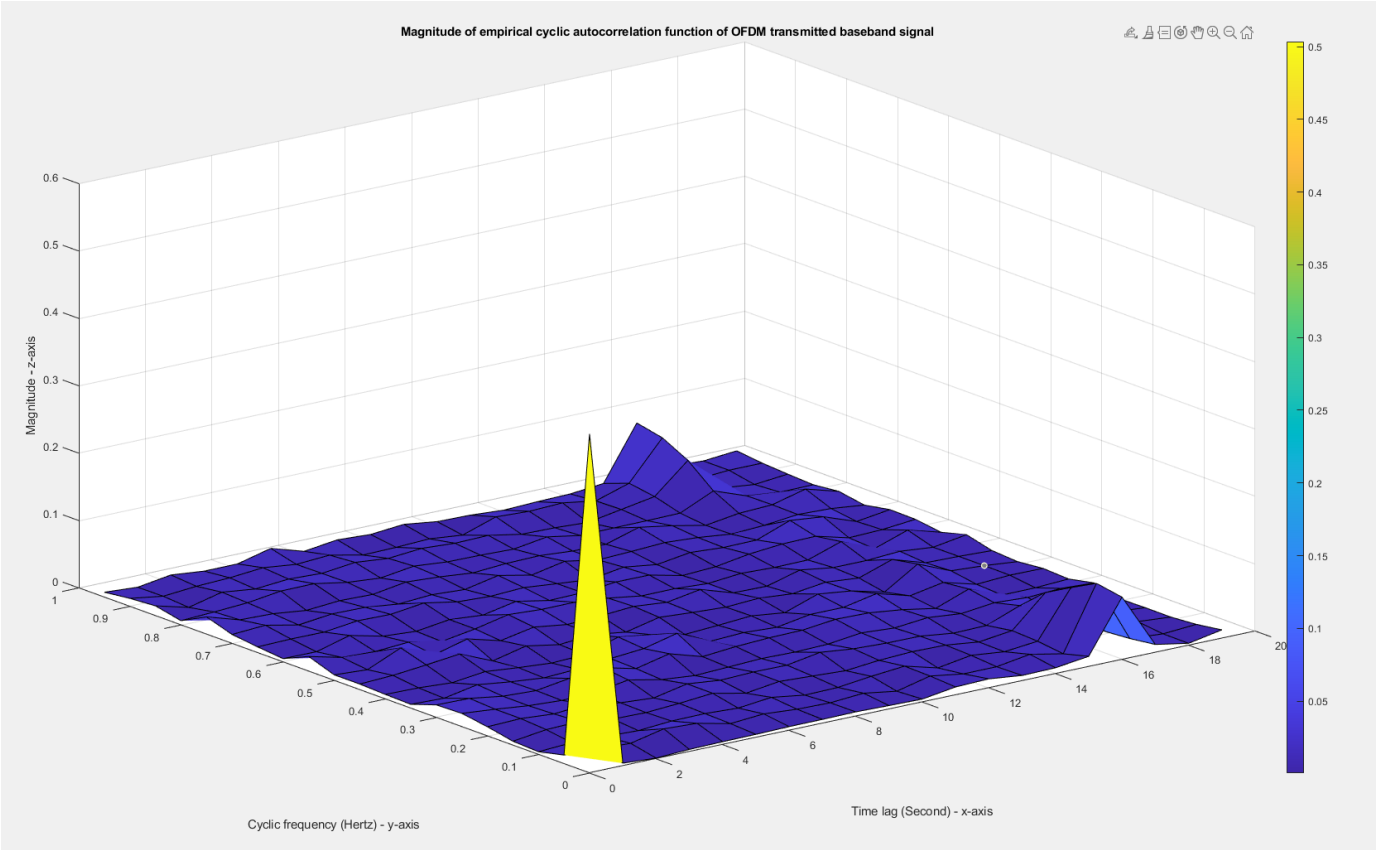

Figure 19: Magnitude of the empirical CAF of an OFDM signal with subcarriers modulated by BPSK (this figure is generated using 100 OFDM symbols, each symbol has 16 subcarriers and 4 cyclic prefixes with the sampling interval 1s)

Outline

Decimated component decomposition:

Proposition (Decimated component decomposition [22])

A DT WSCS process $X[n]$ with a cyclostationarity period N_0 can be decomposed into an N_0-variate stationary process $\mathbf{X}[n]$ consisting of subprocesses $X_i[n] = X[nN_0 + i],\ i \in \{0, 1, \ldots, N_0 - 1\}$.

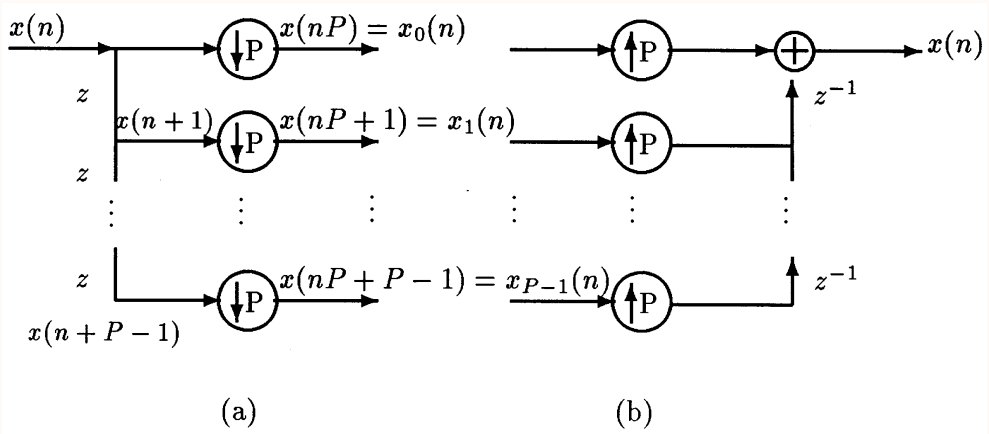

Figure 20: Decimated component decomposition

[22] G. B. Giannakis, "Cyclostationary signal analysis," in *Digital Signal Processing Handbook*, V. K. Madisetti and D. B. Williams, Eds., Boca Raton, FL, USA: CRC Press, 1999, ch. 17