Research Article
Open Access
CrossMark

Sparsification, compression and reconstruction of speech signal data using discrete Rajan transform

Abstract

This paper introduces a novel transform called Discrete Rajan Transform, which could be used to compress speech data during the registration process. The spectrum of the voice data of a registrant is evaluated, sparsed, compressed and then stored in a database. During testing phase, the compressed spectral data is uncompressed, the corresponding original time domain voice data sequence is evaluated from the reconstructed spectrum and then subjected to the verification process. Albeit a number of discrete transforms are available, a special kind of transform called Discrete Rajan Transform (DRT) is preferred to do this job due to its generality and minimal computational complexity. The details of this approach are briefly given in this paper.

Keywords: spectrum, computation, unique matrix, equilibrium segments

Volume 11 Issue 2 - 2025

Prashanthi G,[1] Sathya G,[2] Rajan EG[3]
[1]Data Scientist, Canada
[2]Data Scientist, USA
[3]Chairman, Pentagram Group of Companies, India

Correspondence: EG Rajan, Chairman, Pentagram Group of Companies, India

Received: September 5, 2025 | **Published:** October 10, 2025

Discrete Rajan transform (RT)

Let us consider a discrete signal x(n) of length N. DRT is applied to x(n) and its spectrum X(k) obtained after number of stages where n = $\log_2 N$. At every stage, a unique matrix R_k of dimension $\left(\frac{N}{2^{k-1}} \times \frac{N}{2^{k-1}}\right)$ is constructed as shown below:

$$R_k = \begin{bmatrix} I_p & I_p \\ -e_k^1 . I_p & e_k^1 . I_p \end{bmatrix}$$

where, I refers to p^{th} order identity matrix. For instance, at stage k, one can calculate $p_k = N / 2^k$; $k \in \{1,2,\ldots,n\}$. The term e_k^i is called 'auxiliary information'. e_k^i indicates the inherent relationship between every equilibrium segment of the corresponding signal spectrum generated during the i^{th} stage of computation. Now, let the sequence at a particular stage k be $\bar{s} = \{a,b,c,d,e,f,g,h\}$. Then e_k would indicate certain phasor relationship between the sample points a and e.[1]

$$e_k^i = \begin{cases} -1, & x_k^i(p_k+1) < x_k^i(1) \\ 1, & otherwise \end{cases}$$

where $i = \{1,2,\cdots,2^{k-1}\}$. Let Y_k denote the output sequence at stage k, and it is obtained as

$$Y_k = R_k X_k = \begin{bmatrix} y_k^1 & y_k^2 \ldots & y_k^i \end{bmatrix}$$

Observe that Y_k has $2^k p_k$ elements. When $k = 1$, $X_1 = x$ (the original input sequence) and for $k > 1$, 2^{k-1} equilibrium segments are considered at every stage. Every segment is associated with relevant 'auxiliary information'. e_k refers to the 'auxiliary information' pertaining to equilibrium segments in k stages. Now, the following equations hold:

$$e_k = \begin{bmatrix} e_k^1 & e_k^2 & \cdots & e_k^i \end{bmatrix}$$

For instance, if $s_k = [a\ b\ c\ d\ e\ f\ g\ h]$ and $s_{k+1} = \begin{bmatrix} a & e \\ b & f \\ c & g \\ d & h \end{bmatrix}$ then, e_k^1 and e_k^2 are the auxiliary phasor information pertaining to 'a' and 'e', and 'e' and 'g' respectively. The auxiliary phasor values in e_k are derived as given above. The operator matrix R_k at a stage k is constructed using the values of e_k^1 from e_k. For $k > 1$, the output is restructured into equilibrium segments iteratively, and it is defined as:

$$X_{k+1} = \begin{bmatrix} \bar{x}_{k+1}^1 & \mu_k^1 . \bar{x}_{k+1}^2 \cdots & \mu_k^{i-1} . \bar{x}_{k+1}^i \end{bmatrix}, \text{ where}$$

$$\mu_k = \begin{bmatrix} \mu_k^1 & \mu_k^2 \cdots & \mu_k^{i-1} \end{bmatrix} \text{ and } \mu_k^{i-1} = e_k^1 \times e_k^i \text{ for } k > 1$$

Also,

$$X_{k+1} = \begin{bmatrix} y_k^i(1) & y_k^i(2) \cdots & y_k^i(p_k) \\ y_k^i(p_k+1) & y_k^i(p_k+2) & y_k^i(2p_k) \\ \vdots & \vdots & \vdots \\ y_k^i\left(2^{k-1}p_k+1\right) & \cdots & y_k^i\left(2^k p_k\right) \end{bmatrix}^T$$

$$= \begin{bmatrix} \bar{x}_{k+1}^1 & \bar{x}_{k+1}^2 \cdots & \bar{x}_{k+1}^i \end{bmatrix}$$

Note that X_{k+1} splits signal spectrum into equilibrium segments. This procedure is further continued and finally DRT spectrum obtained after n stages. The first DRT spectral value is known as 'Cumulative Point Index' (CPI), which is essentially a measure of the cumulative energy of the input signal. Consider $x(n) = \{1,9,6,2,3,1,5,1\}$ of length N=8. Now, the DRT computation is carried out in three stages (n=3) so that k = {1, 2, 3}. At the initial stage, k=1, p=4 and $X_1 = x$. At this stage R_1 is of dimension of (8×8). 'Auxiliary information' value of e_1^1 is evaluated as 1. Now, the operating matrix R_1 takes the form

$$R_1 = \begin{bmatrix} I_4 & I_4 \\ -I_4 & I_4 \end{bmatrix}$$

The first stage output is obtained as:

$$Y_1 = R_1 X_1 = \begin{bmatrix} 4\ 10\ 11\ 3\ 2\ -8\ -1\ -1 \end{bmatrix}^T$$

At the second stage k=2, p=2 and R_2 is of dimension (4×4). X_2 is obtained by restructuring Y_1

Thus,

$$\begin{bmatrix} \bar{x}_2^1 & \bar{x}_2^2 \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 10 & -8 \\ 11 & -1 \\ 3 & -1 \end{bmatrix}$$

Two equilibrium segments at this stage could be observed, that is, $i = 2$ and so, $\boldsymbol{e}_2 = \begin{bmatrix} e_2^1 & e_2^2 \end{bmatrix}$. The values are obtained as $\boldsymbol{e}_2 = \begin{bmatrix} 1 & -1 \end{bmatrix}$ and $\boldsymbol{i}_2 = \begin{bmatrix} \mu_2^1 \end{bmatrix}$ and $\mu_2^1 = 1 \times -1 = -1$.

Thus $\boldsymbol{X}_2$ is obtained as

$$\boldsymbol{X}_2 = \begin{bmatrix} 4 & -2 \\ 10 & 8 \\ 11 & 1 \\ 3 & 1 \end{bmatrix}$$

To construct operator matrix $\boldsymbol{R}_2$, $e_2^1 = 1$ is used as:

$$\boldsymbol{R}_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

The output of the second stage is calculated as

$$\mathrm{Y}_2 = \mathrm{R}_2\mathrm{X}_2 = \begin{bmatrix} 15 & -1 \\ 13 & 9 \\ 7 & 3 \\ -7 & -7 \end{bmatrix}$$

For the third and last stage, k=3, p=1 and $\boldsymbol{R}_3$ is of dimension (2×2)

$$\begin{bmatrix} \bar{x}_3^1 & \bar{x}_3^2 & \bar{x}_3^3 & \bar{x}_3^4 \end{bmatrix} = \begin{bmatrix} 15 & 7 & -1 & 3 \\ 13 & -7 & 9 & -7 \end{bmatrix}$$

The values of the 'auxiliary information' are obtained for four equilibrium segments (since $i = 4$) as

$e_3 = \begin{bmatrix} e_3^1 & e_3^2 & e_3^3 & e_3^4 \end{bmatrix} = \begin{bmatrix} -1 & -11 & -1 \end{bmatrix}$

and $\mu_3 = \begin{bmatrix} \mu_3^1 & \mu_3^2 & \mu_3^3 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 \end{bmatrix}$. The value of X_3 is thus calculated as

$$X_3 = \begin{bmatrix} 15 & 7 & 1 & 3 \\ 13 & -7 & -9 & -7 \end{bmatrix}$$

Finally, the DRT spectrum is

$$Y_3 = R_3 X_3 = \begin{bmatrix} 28 & 0 & -8 & -4 \\ 2 & 14 & 10 & 10 \end{bmatrix}$$

The DRT spectrum in the sequence form is X(k) = [28, 2, 0, 14, -8, 10, -4, 10]. Note that the CPI of X(k) is 28.

Inverse discrete Rajan transform (IDRT)

Discrete Rajan Transform (DRT) exhibits both homomorphism and isomorphism properties and in the second case the auxiliary phasor information is preserved. When DRT is viewed as isomorphic function, one would be able to retrieve the original signal data from its DRT spectrum using the inverse transform. Inverse Discrete Rajan Transform (IDRT) is employed to retrieve the input data with the help of e_k^1 and μ_k.Note that DRT operator R_k is obtained using the values of e_k^1 and μ_k. The general expression used to calculate intermediate signal data at each stage is:

$$\tilde{X}_m = \frac{1}{2}\left[R_m Y_m\right] = \begin{bmatrix} x_m^1 & x_m^2 \ldots & x_m^i \end{bmatrix}^T$$

where m = {k, k − 1, ..., 1}. Unlike the forward DRT computation where sequence is split into equilibrium segments, the segments are recombined in the case of IDRT computation to retrieve input sequence iteratively.

When $m = k, Y_m = Y_k$ (DRT output) and for $m < k$,

$$Y_{m-1} = \begin{bmatrix} \bar{Y}_m(1) & \mu_k^1 . \bar{Y}_m(2) \cdots & \mu_k^{i-1} . \bar{Y}_m(i) \end{bmatrix}$$

$$\bar{Y}_{m-1} = \begin{bmatrix} x_m^i(1) & x_m^i(2p_k + 1) \cdots & x_m^i\left(2^{k-1} p_k + 1\right) \\ x_m^i(2) & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ x_m^i(2p_k) & x_m^i\left(2^2 p_k\right) & x_m^i\left(2^k p_k\right) \end{bmatrix}$$

Original input signal is obtained at the last stage of IDRT computation.

Rajan Transform (RT) spectrum of the input sequence x(n) is shown in Tables 1&2.

Table 1 Input sequence x(n) and its RT spectrum X(k)

Index	Data	Stage1	Code1	Stage2	Code2	Stage3	Code3	Stage4	Code4	Stage5	Code5	Spectrum	Code
1	1	34	0	100	0	175	0	378	0	784	0	1596	0
2	2	36	0	104	0	182	0	392	0	812	0	28	0
3	3	38	0	108	0	189	0	406	0	28	0	56	0
4	4	40	0	112	0	196	0	420	0	28	0	0	0
5	5	42	0	116	0	203	0	28	0	56	0	112	0
6	6	44	0	120	0	210	0	28	0	56	0	0	0
7	7	46	0	124	0	217	0	28	0	0	0	0	0
8	8	48	0	128	0	224	0	28	0	0	0	0	0
9	9	50	0	75	0	25	1	54	0	112	0	228	0

Citation: Prashanthi G, Sathya G, Rajan EG. Sparsification, compression and reconstruction of speech signal data using discrete Rajan transform. *Int Rob Auto J.* 2025;11(2):71–80. DOI: 10.15406/iratj.2025.11.00303

Table 1 Continued....

10	10	52	0	78	0	26	1	56	0	116	0	4	0
11	11	54	0	81	0	27	1	58	0	4	0	8	0
12	12	56	0	84	0	28	1	60	0	4	0	0	0
13	13	58	0	87	0	29	1	4	0	8	0	16	0
14	14	60	0	90	0	30	1	4	0	8	0	0	0
15	15	62	0	93	0	31	1	4	0	0	0	0	0
16	16	64	0	96	0	32	1	4	0	0	0	0	0
17	17	66	0	32	0	57	0	118	0	240	0	484	0
18	18	68	0	32	0	58	0	120	0	244	0	4	0
19	19	70	0	32	0	59	0	122	0	4	0	8	0
20	20	72	0	32	0	60	0	124	0	4	0	0	0
21	21	74	0	32	0	61	0	4	0	8	0	16	0
22	22	76	0	32	0	62	0	4	0	8	0	0	0
23	23	78	0	32	0	63	0	4	0	0	0	0	0
24	24	80	0	32	0	64	0	4	0	0	0	0	0
25	25	25	0	25	1	7	1	10	0	16	0	28	0
26	26	26	0	26	1	6	1	8	0	12	0	4	1
27	27	27	0	27	1	5	1	6	0	4	1	8	0
28	28	28	0	28	1	4	1	4	0	4	1	0	0
29	29	29	0	29	1	3	1	4	1	8	0	16	0
30	30	30	0	30	1	2	1	4	1	8	0	0	0
31	31	31	0	31	1	1	1	4	1	0	0	0	0
32	32	32	0	32	1	0	0	4	1	0	0	0	0
33	33	32	0	64	0	121	0	246	0	496	0	996	0
34	34	32	0	64	0	122	0	248	0	500	0	4	0
35	35	32	0	64	0	123	0	250	0	4	0	8	0
36	36	32	0	64	0	124	0	252	0	4	0	0	0
37	37	32	0	64	0	125	0	4	0	8	0	16	0
38	38	32	0	64	0	126	0	4	0	8	0	0	0
39	39	32	0	64	0	127	0	4	0	0	0	0	0
40	40	32	0	64	0	128	0	4	0	0	0	0	0
41	41	32	0	57	0	7	1	10	0	16	0	28	0
42	42	32	0	58	0	6	1	8	0	12	0	4	1
43	43	32	0	59	0	5	1	6	0	4	1	8	0
44	44	32	0	60	0	4	1	4	0	4	1	0	0
45	45	32	0	61	0	3	1	4	1	8	0	16	0
46	46	32	0	62	0	2	1	4	1	8	0	0	0
47	47	32	0	63	0	1	1	4	1	0	0	0	0
48	48	32	0	64	0	0	0	4	1	0	0	0	0
49	49	32	0	0	0	7	0	10	0	16	0	28	0
50	50	32	0	0	0	6	0	8	0	12	0	4	1
51	51	32	0	0	0	5	0	6	0	4	1	8	0
52	52	32	0	0	0	4	0	4	0	4	1	0	0
53	53	32	0	0	0	3	0	4	1	8	0	16	0
54	54	32	0	0	0	2	0	4	1	8	0	0	0
55	55	32	0	0	0	1	0	4	1	0	0	0	0
56	56	32	0	0	0	0	0	4	1	0	0	0	0
57	0	25	1	7	1	7	0	10	0	16	0	28	0
58	0	26	1	6	1	6	0	8	0	12	0	4	1
59	0	27	1	5	1	5	0	6	0	4	1	8	0
60	0	28	1	4	1	4	0	4	0	4	1	0	0
61	0	29	1	3	1	3	0	4	1	8	0	16	0
62	0	30	1	2	1	2	0	4	1	8	0	0	0
63	0	31	1	1	1	1	0	4	1	0	0	0	0
64	0	32	1	0	0	0	0	4	1	0	0	0	0

Inverse Rajan Transform (IRT) of the input spectrum X(k) is shown in Table 2.

Table 2 Input spectrum X(k) and its IRT sequence x(n)

Index	RT Spectrum	Stage1	Stage2	Stage3	Stage4	Stage5	IRT
1	1596	784	378	175	100	34	1
2	28	812	392	182	104	36	2
3	56	28	406	189	108	38	3
4	0	28	420	196	112	40	4
5	112	56	28	203	116	42	5
6	0	56	28	210	120	44	6
7	0	0	28	217	124	46	7
8	0	0	28	224	128	48	8
9	228	112	54	25	75	50	9
10	4	116	56	26	78	52	10
11	8	4	58	27	81	54	11
12	0	4	60	28	84	56	12
13	16	8	4	29	87	58	13
14	0	8	4	30	90	60	14
15	0	0	4	31	93	62	15
16	0	0	4	32	96	64	16
17	484	240	118	57	32	66	17
18	4	244	120	58	32	68	18
19	8	4	122	59	32	70	19
20	0	4	124	60	32	72	20
21	16	8	4	61	32	74	21
22	0	8	4	62	32	76	22
23	0	0	4	63	32	78	23
24	0	0	4	64	32	80	24
25	28	16	10	7	25	25	25
26	4	12	8	6	26	26	26
27	8	4	6	5	27	27	27
28	0	4	4	4	28	28	28
29	16	8	4	3	29	29	29
30	0	8	4	2	30	30	30
31	0	0	4	1	31	31	31
32	0	0	4	0	32	32	32
33	996	496	246	121	64	32	33
34	4	500	248	122	64	32	34
35	8	4	250	123	64	32	35
36	0	4	252	124	64	32	36
37	16	8	4	125	64	32	37
38	0	8	4	126	64	32	38
39	0	0	4	127	64	32	39
40	0	0	4	128	64	32	40
41	28	16	10	7	57	32	41
42	4	12	8	6	58	32	42
43	8	4	6	5	59	32	43
44	0	4	4	4	60	32	44
45	16	8	4	3	61	32	45
46	0	8	4	2	62	32	46
47	0	0	4	1	63	32	47
48	0	0	4	0	64	32	48
49	28	16	10	7	0	32	49
50	4	12	8	6	0	32	50
51	8	4	6	5	0	32	51
52	0	4	4	4	0	32	52
53	16	8	4	3	0	32	53

Citation: Prashanthi G, Sathya G, Rajan EG. Sparsification, compression and reconstruction of speech signal data using discrete Rajan transform. *Int Rob Auto J.* 2025;11(2):71–80. DOI: 10.15406/iratj.2025.11.00303

Table 2 Continued....

54	0	8	4	2	0	32	54
55	0	0	4	1	0	32	55
56	0	0	4	0	0	32	56
57	28	16	10	7	7	25	0
58	4	12	8	6	6	26	0
59	8	4	6	5	5	27	0
60	0	4	4	4	4	28	0
61	16	8	4	3	3	29	0
62	0	8	4	2	2	30	0
63	0	0	4	1	1	31	0
64	0	0	4	0	0	32	0

The advantage of using Rajan Transform for signal processing purposes is that it introduces maximum correlation even in a highly uncorrelated input sequence. For example, a random sequence of length 4096 ranging from values '0' to '255' was generated using a pseudo-random number sequence generator. Figure 1 *shows the random sequence in graphical form.*

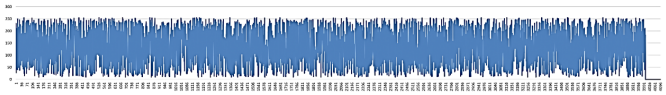

Figure 1 Random number sequence x(n) of length 4096 with values ranging from 0 to 255.

Rajan Transform (RT) is applied to x(n) and RT spectrum X(k) evaluated after 12 stages. Figure 2 shows the steps involved in the evaluation process.

Now, Inverse Rajan Transform (IRT) is applied to the totally ordered spectral sequence X(k) and the original input sequence x(n) is obtained after 12 steps. Refer to Figure 3. Note that the random number sequence is retrieved by IRT as such.[2]

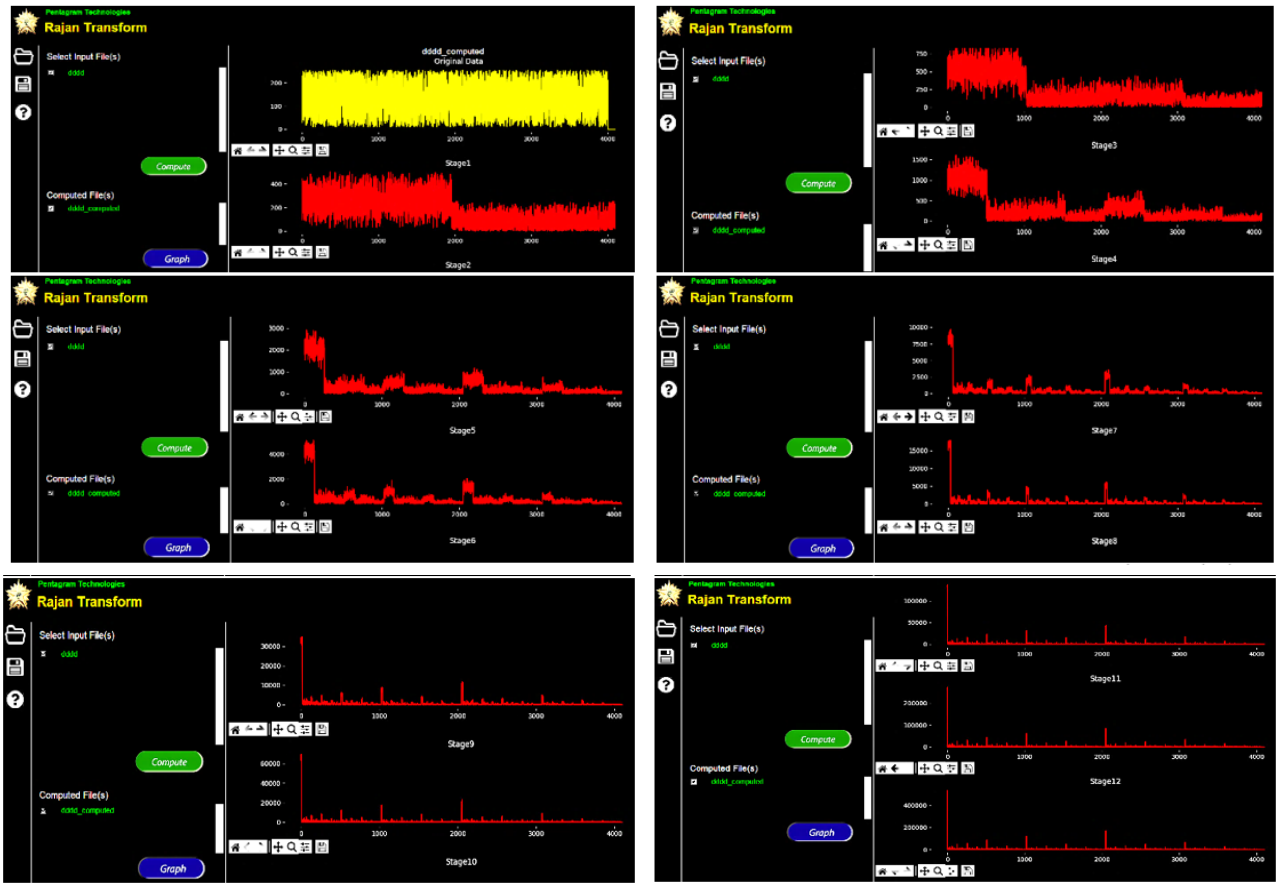

Figure 2 Rajan transform (RT) of the input x(n).

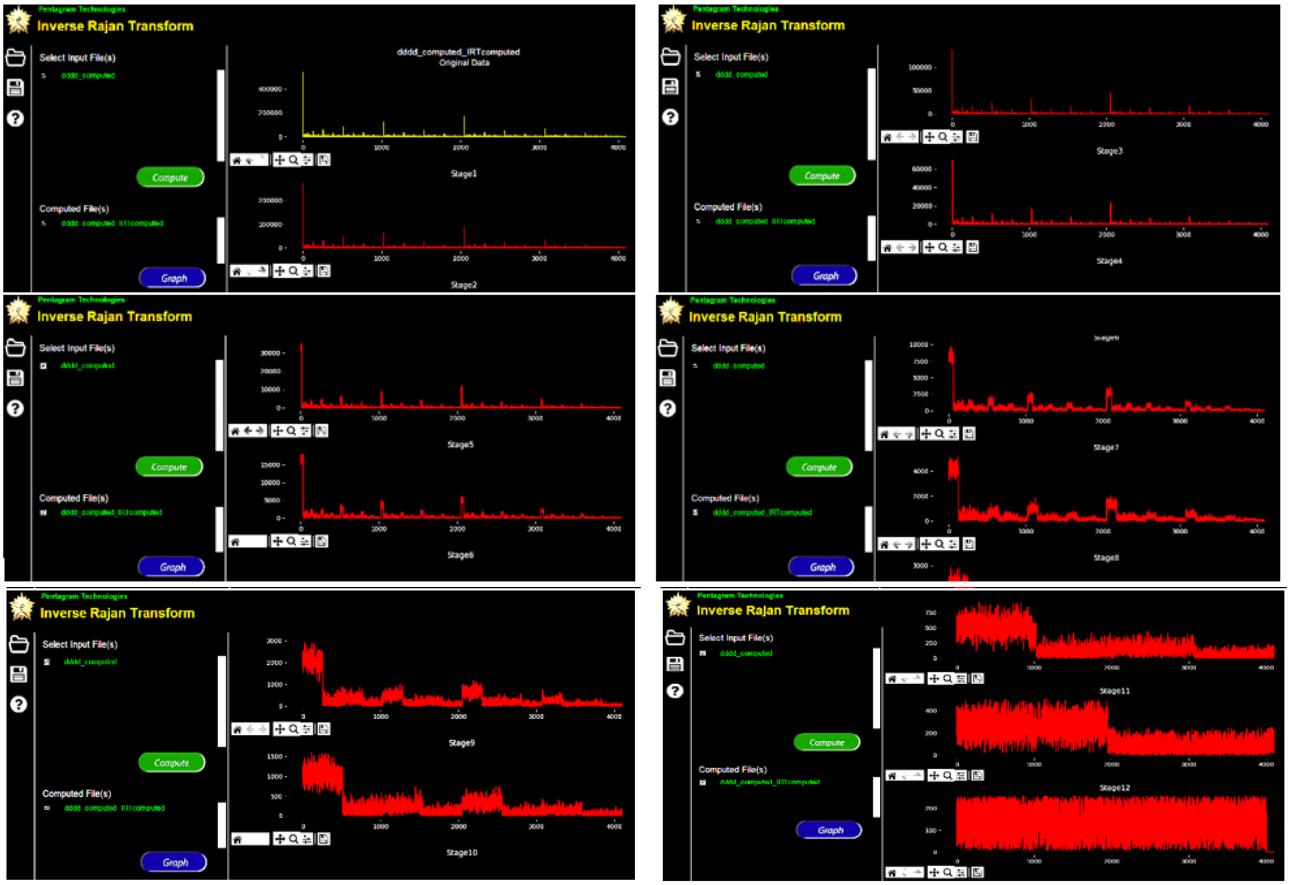

Figure 3 Inverse Rajan transform (IRT) of the spectrum X(k).

Spectral sparsification of speech data using discrete Rajan transform

Let us consider a speech signal sampled at a rate of 16000 Hz. The speech signal of length 48128 samples is segmented into blocks of 8×1 samples and DRT applied to each block. Sparsity is introduced in the data by dropping certain chosen spectral values in the DRT spectrum. A maximum sparsity in the signal is obtained by considering the CPI values alone. This causes considerable loss of information in the signal. Figure 4 shows 6016 blocks of size 8×1 in a speech data of length 48128.

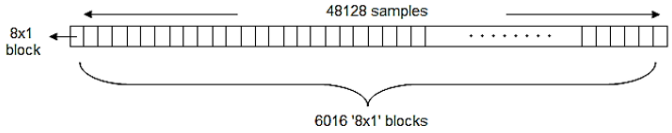

Figure 4 6016 blocks of 8×1 samples.

Since DRT is applied to each block, the resulting DRT spectrum would also consist of 8×1 samples. The first element in every 8×1 block spectrum is its CPI and the remaining its spectral components. Now, sparsing is done in each block spectrum by dropping seven spectral elements other than the CPI. This leads to the compression of the spectrum of voice data of length 48128 to 6016 CPI values. This CPI sequence of length 6016 is stored instead of the original voice data sequence of length 48128. Indeed, one can uncompress by diluting the CPI sequence, that is by introducing 0's in appropriate places and make it as a 0-diluted sequence of length 48128. Then the uncompressed sequence could be decoded using IDRT algorithm. Now the reconstructed voice data sequence is subjected to various procedures meant for speaker identification. In what follows, results due to three case studies are presented.

Case #1: Let us consider a sample real time speech data discrete sequence $\mathbf{x(n)}$ of length 64.

$\mathbf{x(n)}$ = 0.123016357, 0.137512207, 0.163513184, 0.169403076, 0.154754639, 0.151794434, 0.146972656, 0.156585693, 0.148101807, 0.143676758, 0.13269043, 0.125610352, 0.11920166, 0.116485596, 0.11630249, 0.09942627, 0.09197998, 0.091918945, 0.081085205, 0.05847168, 0.048126221, 0.035888672, 0.028137207, 0.014251709, 0.006439209, -0.001617432, -0.034362793, -0.049682617, -0.065917969,

-0.080383301, -0.110229492, -0.137573242, -0.155883789, -0.176696777, -0.194366455, -0.22088623,

-0.228210449, -0.254638672, -0.269989014, -0.274810791, -0.277496338, -0.258209229, -0.270233154,

-0.259857178, -0.242462158, -0.221923828, -0.206390381, -0.176635742, -0.140960693, -0.12878418,

-0.109222412, -0.099914551, -0.063812256, -0.037109375, -0.00189209, 0.032470703, 0.058349609, 0.06451416, 0.072418213, 0.092376709, 0.099212646, 0.116790771, 0.114685059, 0.119567871.

Citation: Prashanthi G, Sathya G, Rajan EG. Sparsification, compression and reconstruction of speech signal data using discrete Rajan transform. *Int Rob Auto J.* 2025;11(2):71–80. DOI: 10.15406/iratj.2025.11.00303

Figure 5 shows the plot of $x(n)$.

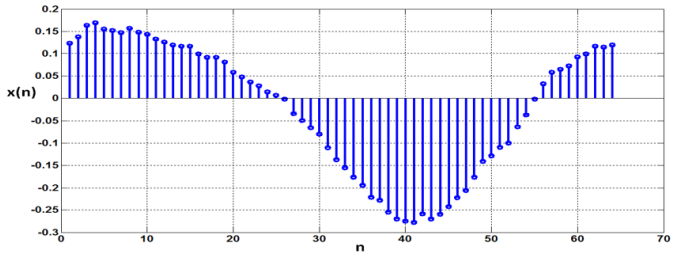

Figure 5 Plot of x(n).

DRT is applied to $x(n)$ in block-wise manner and corresponding spectral blocks obtained as $X(k)$.

$X(k)$ = 1.203552, -0.02704, -0.0694, 0.003967, -0.01666, -0.01373, -0.07538, -0.02118, 1.001495, 0.031097, 0.053436, -0.01682, 0.098663, -0.00809, 0.013519, 0.011505, 0.44986, 0.048798, 0.085968, -0.0242, 0.197052,

-0.00345, 0.002716, -0.0209, -0.47333, 0.065186, 0.190369, -0.02014, 0.31488, -0.01843, -0.01263, 0.005615,

-1.77548, 0.078583, 0.144623, 0.0159, 0.279816, 0.016083, 0.020721, -0.02731, -1.91321,-0.07996, -0.08698, 0.000305, -0.21838, 0.02063, 0.075745, -0.01813, -0.54922, -0.08255, -0.19211, 0.004791, -0.40854, 0.039581, 0.070892, -0.01053, 0.737915, -0.04858, -0.06018, 0.001099, -0.1626, -0.00366, -0.02368, 0.026489.

Now DRT spectrum X(k) is sparsified as follows. $X(k)$ is sparsed by retaining the CPI alone in every block of length 8 and making all other elements '0'. Then, the sparsed spectral sequence is $X_{s1}(k)$ = 1.203552246, 0, 0, 0, 0, 0, 0, 0, 1.001495361, 0, 0, 0, 0, 0, 0, 0, 0.449859619, 0, 0, 0, 0, 0, 0, 0, -0.473327637, 0, 0, 0, 0, 0, 0, 0, -1.775482178, 0, 0, 0, 0, 0, 0, 0, -1.913208008, 0, 0, 0, 0, 0, 0, 0, -0.549224854, 0, 0, 0, 0, 0, 0, 0, 0.737915039, 0, 0, 0, 0, 0, 0, 0.

Note that the sparsed spectral sequence has just 8 nonzero samples. The compressed version of X_s1 (k) is X_s1^' (k)= 1.203552246, 1.001495361, 0.449859619, -0.473327637, -1.775482178, -1.913208008,

-0.549224854, 0.737915039. X_s1^' (k)is stored as the representative biometric vector corresponding to a speaker. The degree of sparsity obtained in this case is 12.5%. Figure 6 shows the plot of X(k). Figure 7 shows the plot of X_s1 (k). Figure 8 shows compressed form of the sparsed spectral sequence.

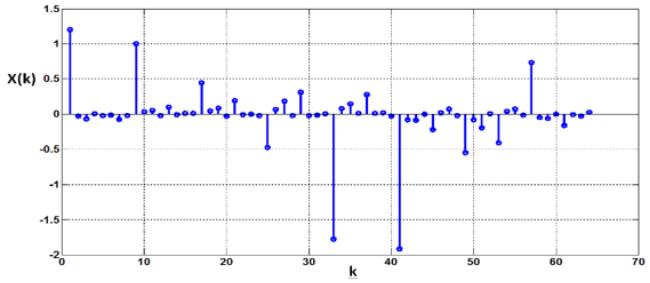

Figure 6 Plot of X(k), the spectrum of x(n).

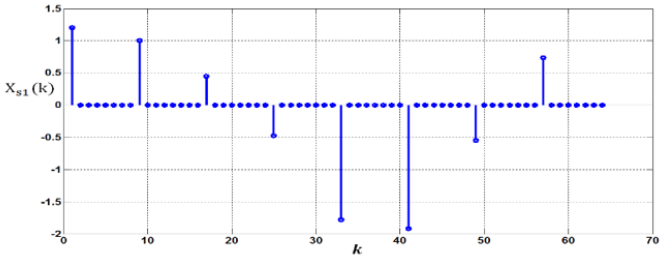

Figure 7 Plot of $X_{s1}(k)$ the sparsed spectral sequence.

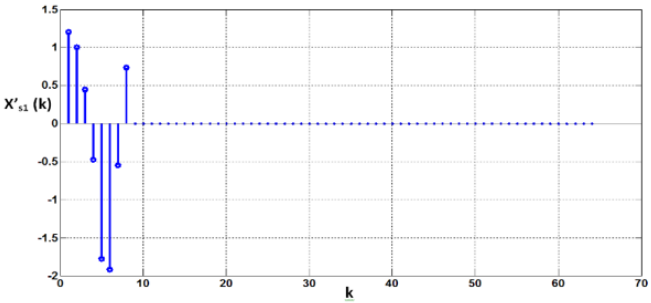

Figure 8 Compressed form of sparsed sequence.

During the testing phase, $X'_{s1}(k)$ is uncompressed as $X_{s1}(k)$. Now, IDRT is applied to $X_{s1}(k)$ and $[\tilde{x}(n)]_1$is obtained as the reconstructed form of the original signal. $[\tilde{x}(n)]_1$= 0.150444031, 0.150444031, 0.150444031, 0.150444031, 0.150444031, 0.150444031, 0.150444031, 0.150444031, 0.12518692, 0.12518692, 0.12518692, 0.12518692, 0.12518692, 0.12518692, 0.12518692, 0.12518692, 0.056232452, 0.056232452, 0.056232452, 0.056232452, 0.056232452, 0.056232452, 0.056232452, 0.056232452, -0.059165955, -0.059165955,

-0.059165955, -0.059165955, -0.059165955, -0.059165955, -0.059165955, -0.059165955, -0.221935272,

-0.221935272, -0.221935272, -0.221935272, -0.221935272, -0.221935272, 0.221935272,

-0.221935272, -0.239151001, -0.239151001, -0.239151001, -0.239151001, -0.239151001, -0.239151001,

-0.239151001, -0.239151001, -0.068653107, -0.068653107, -0.068653107, -0.068653107, -0.068653107,

-0.068653107, -0.068653107, -0.068653107, 0.09223938, 0.09223938, 0.09223938, 0.09223938, 0.09223938, 0.09223938, 0.09223938, 0.09223938.

Figure 9 shows the plot of $[\tilde{x}(n)]_1$. Figure 10 shows x(n) and $[\tilde{x}(n)]_1$ in the same plot.

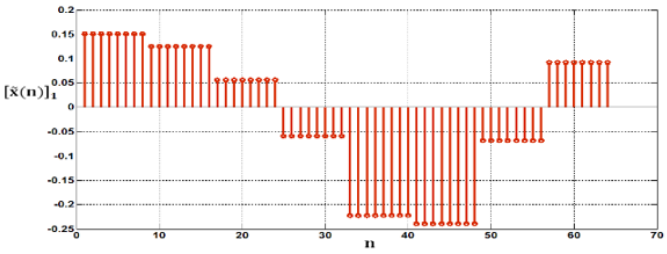

Figure 9 Plot of $[\tilde{x}(n)]_1$ the reconstructed sequence.

Citation: Prashanthi G, Sathya G, Rajan EG. Sparsification, compression and reconstruction of speech signal data using discrete Rajan transform. *Int Rob Auto J.* 2025;11(2):71–80. DOI: 10.15406/iratj.2025.11.00303

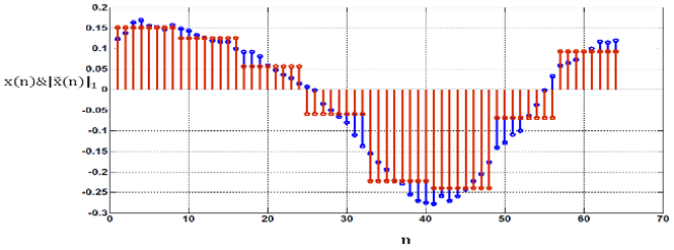

Figure 10 Plot of x(n) and $[\tilde{x}(n)]_1$.

The error sequence $x(n) - [\tilde{x}(n)]_1$ is evaluated and presented below.

$x(n) \sim [\tilde{x}(n)]_1$ = -0.027427673, -0.012931824, 0.013069153, 0.018959045, 0.004310608, 0.001350403,

-0.003471375, 0.006141663, 0.022914886, 0.018489838, 0.00750351, 0.000423431, -0.00598526,

-0.008701324, -0.00888443, -0.025760651, 0.035747528, 0.035686493, 0.024852753, 0.002239227,

-0.008106232 -0.020343781, -0.028095245, -0.041980743, 0.065605164, 0.057548523, 0.024803162, 0.009483337, -0.006752014, -0.021217346, -0.051063538, -0.078407288, 0.066051483, 0.045238495, 0.027568817, 0.001049042, -0.006275177, -0.0327034, -0.048053741, -0.052875519, -0.038345337,

-0.019058228, -0.031082153, -0.020706177, -0.003311157, 0.017227173, 0.03276062, 0.062515259,

-0.072307587, -0.060131073, -0.040569305, -0.031261444, 0.004840851, 0.031543732, 0.066761017, 0.101123810, -0.033889771, -0.02772522, -0.019821167, 0.000137329, 0.006973267, 0.024551392, 0.022445679, 0.027328491. Now the Error Dynamic Range is evaluated as:

$$\text{maximum error} - \text{minimum error} = 0.101123810 - (-0.078407288) = 0.179531098.$$

The error plot $x(n) - [\tilde{x}(n)]_1$ is shown in Figure 11.

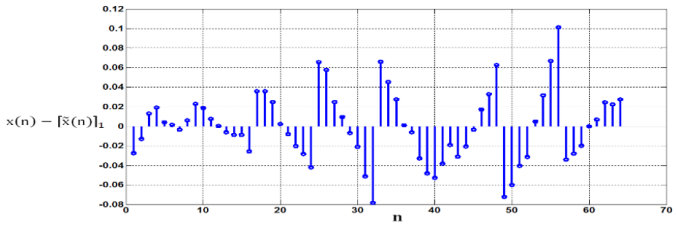

Figure 11 Plot of Error Dynamic Range (EDR).

Case #2: Let us consider the same sample data sequence $x(n)$ of length 64 used in Case #1. Now $X(k)$ is sparsed keeping the CPI and the mid frequency component in every block of length 8 and forcing other six elements to 0. Then, the sparsed sequence is obtained as $X_{s2}(k)$ = 1.203552246, 0, 0, 0, 0.016662598, 0, 0, 0, 1.001495361, 0, 0, 0, 0.09866333,0, 0, 0, 0.449859619, 0, 0, 0, 0.197052002, 0, 0, 0, -0.473327637, 0, 0, 0, 0.314880371, 0, 0, 0, -1.775482178, 0, 0, 0, 0.279815674, 0, 0, 0, -1.913208008, 0, 0, 0, -0.218383789, 0, 0, 0,-0.549224854, 0, 0, 0, -0.408538818, 0, 0, 0, 0.737915039, 0, 0, 0, -0.162597656, 0, 0, 0.

Figure 12 shows the plot of $X_{s2}(k)$. The sparsed spectral representation in this case has just 16 nonzero samples. The compressed version of $X_{s2}(k)$ is obtained as:

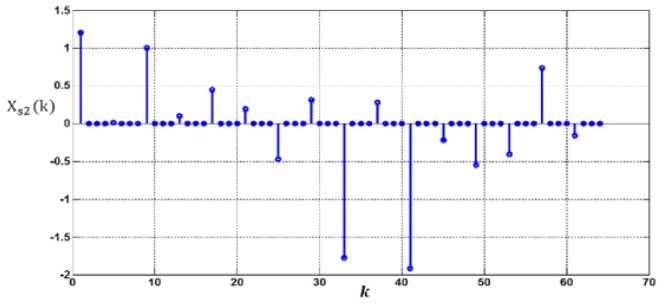

Figure 12 Plot of $X_{s2}(k)$ the sparsed sequence.

$X'_{s2}(k)$= 1.203552246, 0.016662598, 1.001495361, 0.09866333, 0.449859619, 0.197052002, -0.473327637, 0.314880371, -1.775482178, 0.279815674, -1.913208008, -0.218383789, -0.549224854, -0.408538818, 0.737915039, -0.162597656.

Figure 13 shows the plot of the compressed form $X'_{s2}(k)$ of sparsed spectral sequence $X_{s2}(k)$.

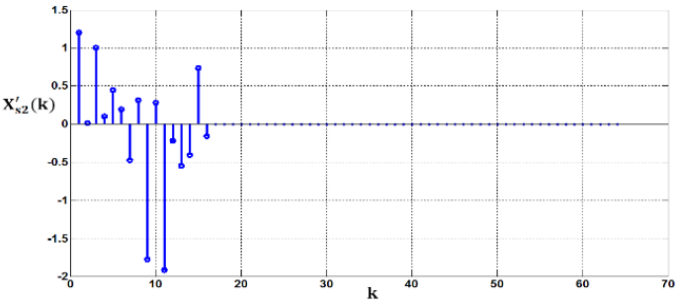

Figure 13 Compressed form of sparsed sequence.

$X'_{s2}(k)$ is stored as a representative biometric vector of a speaker. The degree of sparsity obtained in this case is 25%. The compressed sequence is shown in Figure 13. During the testing phase, $X'_{s2}(k)$ is uncompressed and $X_{s2}(k)$ obtained. Now, IDRT is applied to $X_{s2}(k)$ and the reconstructed form of the original signal is obtained as $[\tilde{x}(n)]_2$

$[\tilde{x}(n)]_2$=0.148361206,0.148361206,0.148361206,0.148361206, 0.152526855, 0.152526855, 0.152526855, 0.152526855, 0.137519836, 0.137519836, 0.137519836, 0.137519836, 0.112854004, 0.112854004, 0.112854004, 0.112854004, 0.080863953, 0.080863953, 0.080863953, 0.080863953, 0.031600952, 0.031600952, 0.031600952, 0.031600952, -0.019805908, -0.019805908, -0.019805908, -0.019805908,

-0.098526001, -0.098526001, -0.098526001, -0.098526001, -0.186958313, -0.186958313, -0.186958313,

-0.186958313, -0.256912231, -0.256912231, -0.256912231, -0.256912231, -0.266448975, -0.266448975,

-0.266448975, -0.266448975, -0.211853027, -0.211853027, -0.211853027, -0.211853027, -0.119720459,

-0.119720459, -0.119720459, -0.119720459, -0.017585754, -0.017585754, -0.017585754, -0.017585754, 0.071914673, 0.071914673, 0.071914673, 0.071914673, 0.112564087, 0.112564087, 0.112564087, 0.112564087.

Citation: Prashanthi G, Sathya G, Rajan EG. Sparsification, compression and reconstruction of speech signal data using discrete Rajan transform. *Int Rob Auto J.* 2025;11(2):71–80. DOI: 10.15406/iratj.2025.11.00303

Figure 14 shows the plot of $[\tilde{x}(n)]_2$ Figure 15 shows x(n) and $[\tilde{x}(n)]_2$ in the same plot.

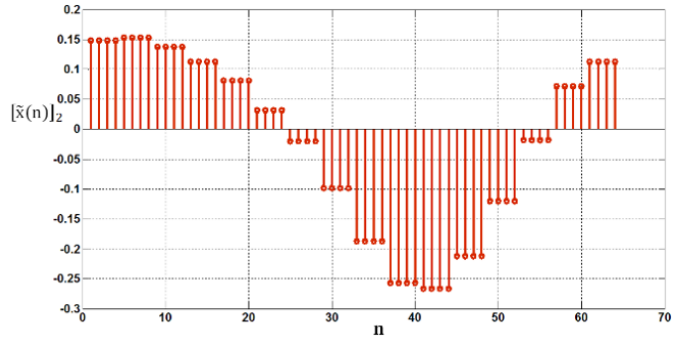

Figure 14 Plot of $[\tilde{x}(n)]_2$ reconstructed sequence

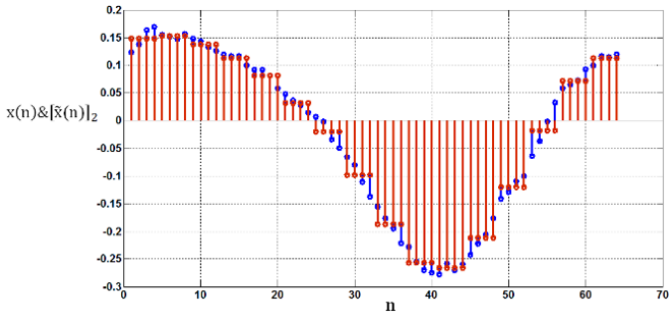

Figure 15 Plot of x(n) and $[\tilde{x}(n)]_2$.

The error sequence $x(n) - [\tilde{x}(n)]_2$ is evaluated and presented below.

$x(n) - [\tilde{x}(n)]_2$ = -0.02534, -0.01085, 0.015152, 0.021042, 0.002228, -0.00073, -0.00555, 0.004059, 0.010582, 0.006157, -0.00483, -0.01191, 0.006348, 0.003632, 0.003448, -0.01343, 0.011116, 0.011055, 0.000221, -0.02239, 0.016525, 0.004288, -0.00346, -0.01735, 0.026245, 0.018188, -0.01456, -0.02988, 0.032608, 0.018143, -0.0117, -0.03905, 0.031075, 0.010262, -0.00741, -0.03393, 0.028702, 0.002274, -0.01308, -0.0179, -0.01105, 0.00824, -0.00378, 0.006592, -0.03061, -0.01007, 0.005463, 0.035217, -0.02124, -0.00906, 0.010498, 0.019806, -0.04623, -0.01952, 0.015694, 0.050056, -0.01357, -0.0074, 0.000504, 0.020462, -0.01335, 0.004227, 0.002121, 0.007004.

Now the Error Dynamic Range (EDR) is evaluated as

$$\text{maximum error} - \text{minimum error} = 0.050056 - (-0.046230) = 0.096286.$$

The error plot $x(n) - [\tilde{x}(n)]_2$ is shown in Figure 16.

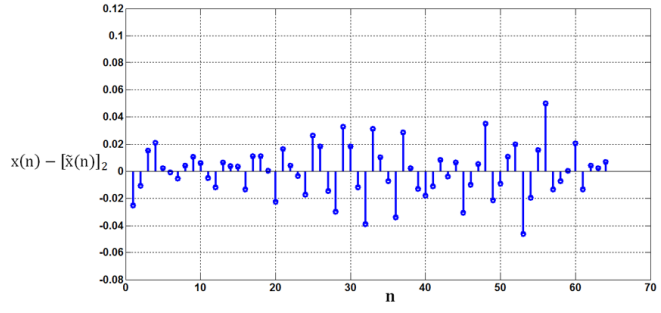

Figure 16 Plot of Error Dynamic Range (EDR).

One can visualize from Figure 16 that the reconstructed discrete sequence of a speaker $[\tilde{x}(n)]_1$ from the sparsified data keeping the CPI alone in every block of length 8 and dropping other elements, that is $X_{s1}(k)$, is close to the original sequence $x(n)$. One can also see from Figure 15 that the reconstructed discrete sequence of a speaker $[\tilde{x}(n)]_2$ from the sparsified data keeping the CPI and the mid frequency component in every block of length 8 and dropping others, that is $X_{s2}(k)$, is very much closer to the original sequence $x(n)$.

Note:

Error Dynamic Range (EDR) due to sparsing of data to 12.5% of voice data is 0.179531098.

Error Dynamic Range (EDR) due to sparsing of data to 25% of voice data is 0.096286000

In addition, DRT algorithm introduces a computational complexity of $\log_N N$. Both forward and inverse computation would then introduce a total complexity of $2\log_N N$.

The above observations have been confirmed by experimenting with TIMIT (Texas Instruments-Massachusetts Institute of Technology) database and a self-collected database. Now, let us take for instance, the case of a real time voice data x(n) of length 48128. Figure 17 shows one such data.

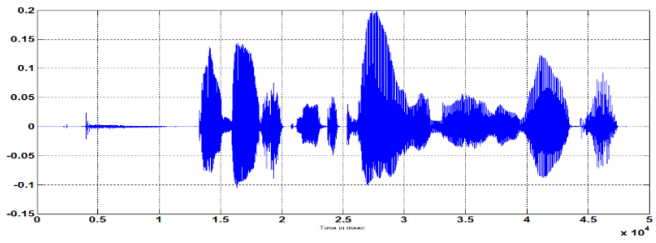

Figure 17 Original speech waveform.

DRT is applied to the x(n) and its spectrum X(k) obtained. Now X(k) is sparsed by retaining only CPI values of all the 6016 blocks. Now, 6016 spectral values would occupy 12.5% of the actual memory allotted to accommodate 48128 samples. Figure 18 shows $[\tilde{x}(n)]_1$, the voice data reconstructed from 12.5% spectral data. Similarly, X(k) is sparsed by retaining CPI values and mid frequency components of all the 6016 blocks. Now, 12032 spectral values occupy 25% of the actual memory allotted to accommodate 48128 samples. Figure 19 shows x(n) and $[\tilde{x}(n)]_1$ presented in the same plot.

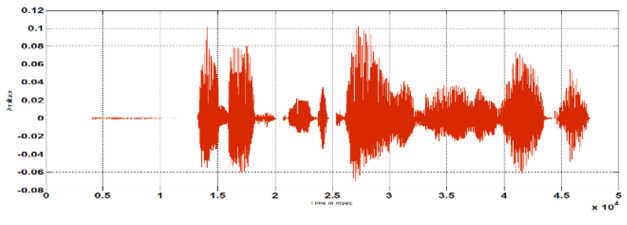

Figure 18 Waveform reconstructed from 12.5% spectral data.

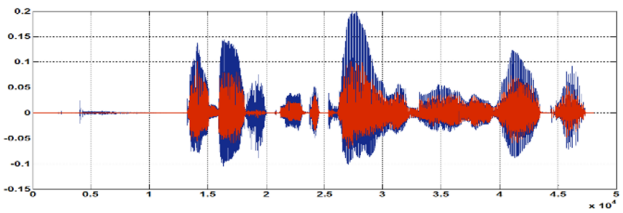

Figure 19 Plot of x(n) and $[\tilde{x}(n)]_1$.

Citation: Prashanthi G, Sathya G, Rajan EG. Sparsification, compression and reconstruction of speech signal data using discrete Rajan transform. *Int Rob Auto J.* 2025;11(2):71–80. DOI: 10.15406/iratj.2025.11.00303

Figure 20 shows $[\tilde{x}(n)]_2$, the voice data reconstructed from 25% spectral data. Figure 21 shows x(n) and $[\tilde{x}(n)]_2$ presented in the same plot.

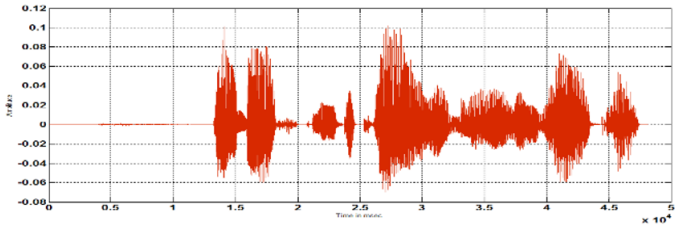

Figure 20 Waveform reconstructed from 25% spectral data.

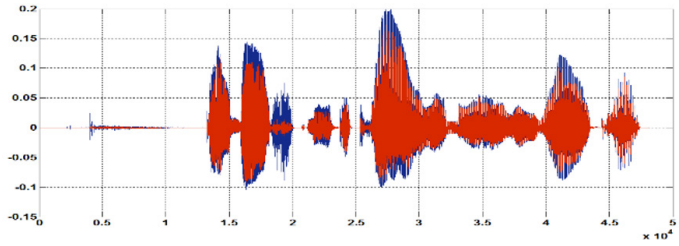

Figure 21 Plot of x(n) and $[\tilde{x}(n)]_2$.

One can visualize from Figures 18 & 19 that the reconstructed voice signal $[\tilde{x}(n)]_1$ of a speaker from the sparsified data keeping the CPI alone in every block of length 8 and dropping other elements, that is $X_{s1}(k)$, is close to the original signal $x(n)$. One can also see from Figures 20 & 21 that the reconstructed voice signal $[\tilde{x}(n)]_2$ of a speaker from the sparsified data keeping the CPI and the mid frequency component in every block of length 8 and dropping other elements to 0, that is $X_{s2}(k)$, is very much closer to the original signal $x(n)$.

Conclusions

To reduce data size, the data compression is performed by sparsifying speech data using Temporal Sparsing and Spectral Domain Sparsing. Temporal Sparsing advocates uniform sparsing of data and the resulting sequence is compressed to 50% of the original data. Alternatively, if Spectral Domain Sparsing is performed, the resulting sequence is further compressed, the database is reduced to 25% or 12.5% of its original size. This reduction in database size indicates a huge data reduction. Sparsification, Data Compression and Reconstruction using Discrete Rajan Transform has been proposed in this paper due to its generality and minimal computational complexity. Error Dynamic Range (EDR) due to sparsing of data to 25% of voice data is very small when compared to the EDR due to sparsing of data to 12.5% of voice data and hence the former is better.

Acknowledgements

None.

Conflicts of interest

Authors declares that there is no conflict of interest.

References

1. Prashanthi G. Signal sparsification with discrete Rajan transform (DRT): principles, properties and applications, MS Thesis submitted to the faculty of computing, engineering and technology, Staffordshire University; 2012.
2. Prashanthi G, Satyanand Singh, Rajan EG, et al. Sparsification of voice data using discrete rajan transform and its applications in speaker recognition. IEEE International Conference on Systems, Man, and Cybernetics October 5-8, 2014, San Diego, CA, USA.