Opulent Voice Numerology Study

Abraxas3d

July 2022

Original Quadratic Permutation Polynomial From M17 Protocol

f(x) = 45x^2 + 92x with length 368

Treating this as Z/368Z

we need a new f(x) for Opulent Voice frame length of 1480

```
x = (0:367)
```

```
x = 1×368
     0     1     2     3     4     5     6     7     8     9    10    11    12 ···
```

```
permutation = mod(45*x + 92.*x.*x, 368)
```

```
permutation = 1×368
     0   137    90   227   180   317   270    39   360   129    82   219   172 ···
```

```
permutation_reversed = mod(45*permutation + 92.*permutation.*permutation, 368)
```

```
permutation_reversed = 1×368
     0     1     2     3     4     5     6     7     8     9    10    11    12 ···
```

```
if length(permutation) == length(unique(permutation))
    disp('There are no repeats in original vector')

end
```

```
There are no repeats in original vector
```

```
for j = 1:367
    original_distance(j) = abs(permutation(j) - permutation(j+1));
end
    disp('Minimum distance is:')
```

```
Minimum distance is:
```

```
    min(original_distance)
```

```
ans = 47
```

```
    disp('This is')
```

```
This is
```

```
    (min(original_distance)/368)*0.04
```

```
ans = 0.0051
```

```
    disp('milliseconds.')
```

```
milliseconds.
```

```

```

New Quadratic Permutation Polynomial for Opulent Voice

Maximum spread quadratic permutation polynomial example.

```
k = (6)
```

```
k = 6
```

```

linear_term = 2.^k - 1
```

```
linear_term = 63
```

```
quadratic_term = 2.^(k+1)
```

```
quadratic_term = 128
```

```
modulo = 2.^(2.*k -1)
```

```
modulo = 2048
```

```

max_spread = mod(linear_term*x + quadratic_term.*x.*x, modulo)
```

```
max_spread = 1×368
           0         191         638        1341         252        1467 ···
```

```


if length(max_spread) == length(unique(max_spread))
    disp('There are no repeats in permutated vector')

end
```

```
There are no repeats in permutated vector
```

```

```

Determine new coefficients for frame length of 1480

```
368/4
```

```
ans = 92
```

```
1480/4
```

```
ans = 370
```

```
92/2
```

```
ans = 46
```

```
370/2
```

```
ans = 185
```

```
opv = (0:1479) %index of bit locations for a frame
```

```
opv = 1×1480
     0     1     2     3     4     5     6     7     8     9    10    11    12 ···
```

```
opv_permutation = mod(177*opv + 370.*opv.*opv, 1480) %experimentally determined the lin
```

```
opv_permutation = 1×1480
           0         547         354         901         708        1255 ···
```

```
%test for reversibility
opv_permutation_reversed = mod(177*opv_permutation + 370.*opv_permutation.*opv_permutat
```

```
opv_permutation_reversed = 1×1480
           0         989         498           7         996         505 ···
```

```
%test for complete coverage
if length(opv_permutation) == length(unique(opv_permutation))
    disp('There are no repeats in opv_permutation vector')
else
    disp('Problem with this one')
    errors = unique(opv_permutation)

end
```

```
There are no repeats in opv_permutation vector
```

```
%test for minimum distance between adjacent bits
for i = 1:1479
    distance(i) = abs(opv_permutation(i) - opv_permutation(i+1));
end
    disp('Adjacent bits have a ')
```

```
Adjacent bits have a
```

```
    minimum_distance = min(distance)
```

```
minimum_distance = 193
```

```
    disp('This resuls in a')
```

```
This resuls in a
```

```
    minimum_time = (min(distance)/1480)*0.04
```

```
minimum_time = 0.0052
```

```
    disp('milliseconds.')
```

```
milliseconds.
```

```
    disp('This is ')
```

```
This is
```

```
    minimum_time*1000
```

```
ans = 5.2162
```

```
    disp("microseconds.")
```

```
microseconds.
```

```
%test for maximum distance between adjacent bits
for i = 1:1479
    distance(i) = abs(opv_permutation(i) - opv_permutation(i+1));
end
    disp('Adjacent bits have a ')
```

```
Adjacent bits have a
```

```
    maximum_distance = max(distance)
```

```
maximum_distance = 1287
```

Randomizer Table

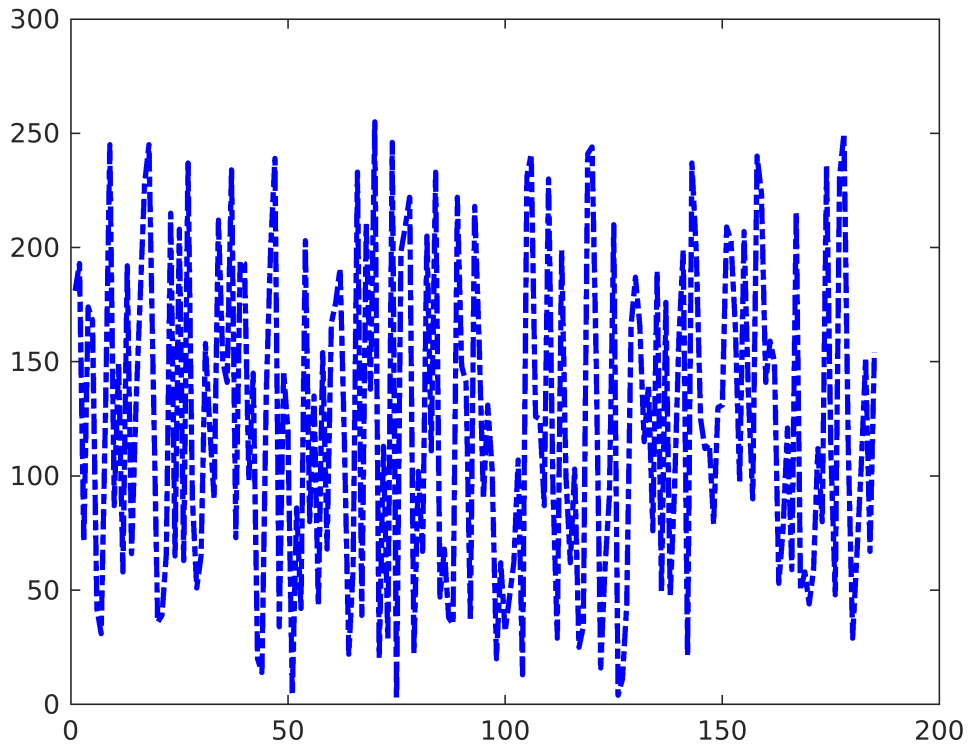
Select 185 random numbers (1 to 255) from a uniform distribution.

```
pd = makedist('Uniform'); % Uniform distribution
pd_array = random(pd,1480/8,1);
r = ceil(255.*pd_array);
r_hex = dec2hex(r);


x = 1:1480/8;

figure
plot(x,r,'b-.','LineWidth',2)
```

Warning: MATLAB has disabled some advanced graphics rendering features by switching to software OpenGL. For more information, click here.



Selected Randomizer Digits in Hexadecimal

'A3' '81' '5C' 'C4' 'C9' '08' '0E' '53' 'CC' 'A1' 'FB' '29' '9E' '4F' '16' 'E0' '97' '4E'
'2B' '57' '12' 'A7' '3F' 'C2' '4D' '6B' '0F' '08' '30' '46' '11' '56' '0D' '1A' '13' 'E7'
'50' '97' '61' 'F3' 'BE' 'E3' '99' 'B0' '64' '39' '22' '2C' 'F0' '09' 'E1' '86' 'CF' '73'
'59' 'C2' '5C' '8E' 'E3' 'D7' '3F' '70' 'D4' '27' 'C2' 'E0' '81' '92' 'DA' 'FC' 'CA' '5A'
'80' '42' '83' '15' '0F' 'A2' '9E' '15' '9C' '8B' 'DB' 'A4' '46' '1C' '10' '9F' 'B3' '47'
'6C' '5E' '15' '12' '1F' 'AD' '38' '3D' '03' 'BA' '90' '8D' 'BE' 'D3' '65' '23' '32' 'B8'
'AB' '10' '62' '7E' 'C6' '26' '7C' '13' 'C9' '65' '3D' '15' '15' 'ED' '35' 'F4' '57' 'F5'
'58' '11' '9D' '8E' 'E8' '34' 'C9' '59' 'F8' 'D6' 'B6' '37' '04' '36' '89' '1C' 'DA' 'E9'
'56' '78' '01' '50' '7C' '43' 'AF' 'E9' '92' '44' 'ED' '11' 'A0' 'F2' '84' 'F4' '46' '87' 'E9'
'37' 'D3' '24' '70' 'E0' 'B4' '7F' '9C' '14' '3E' '07' 'D8' '04' '8D' '1F' '96' '9F' 'BF' '50'
'EA' 'C8' '1A'

Selected Randomizer Digits in Decimal Format

163 129 92 196 201 8 14 83 204 161 251 41 158 79 22 224 151 78 43 87 18 167 63 194 77 107 15 8 48 70 17 86 13 26 19 231 80 151 97 243 190 227 153 176 100 57 34 44 240 9 225 134 207 115 89 194 92 142 227 215 63 112 212 39 194 224 129 146 218 252 202 90 128 66 131 21 15 162 158 21 156 139 219 164 70 28 16 159 179 71 108 94 21 18 31 173 56 61 3 186 144 141 190 211 101 35 50 184 171 16 98 126 198 38 124 19 201 101 61 21 21 237 53 244 87 245 88 17 157 142 232 52 201 89 248 214 182 55 4 54 137 28 218 233 86 120 1 80 124 67 175 233 146 68 237 17 160 242 132 244 70 135 233 55 211 36 112 224 180 127 156 20 62 7 216 4 141 31 150 159 191 80 234 200 26