Algorithm 1 SimLocRF Dataset Generation in MATLAB - Part 1

```
1. Initialize Parameters
1.1. RF Environment Parameters
  parameters.f                                   ▷ Operating frequency
  parameters.c                                   ▷ Speed of light
  parameters.lambda                              ▷ Wavelength
  parameters.Pt                                  ▷ Transmit power
  parameters.Gt, parameters.Gr                   ▷ Transmit and Receive antenna gain
1.2. RF Configuration Parameters
  parameters.num_antennas                        ▷ Number of antennas
  parameters.num_angles                          ▷ Number of angles
  parameters.d_s, parameters.theta_s             ▷ Source coordinates
  parameters.d_r, parameters.theta_r             ▷ Receivers coordinates
1.3. Data Resolution and Quality Parameters
  parameters.snr_level                           ▷ Signal-to-Noise ratio
  parameters.resolution                          ▷ Angular resolution 1, 10 in degrees
  parameters.num_instances_per_angle             ▷ Data points per angle
2. Generate Dataset
  [features, labels] = generateDataset(parameters)
2.1.1. Generate Features as Powers
  features_Powers = zeros(num_angles * num_instances_per_angle,
num_antennas)
  i = 1:parameters.num_angles
    n = 1:parameters.num_instances_per_angle
      j = 1:length(received_Power)               ▷ received_Power from Friis
equation
        received_Signal = generateSignal(received_Power, f)
        noisy_received_Signal = addNoise(received_Signal,
snr_level)
                                                 ▷ Add noise from AWGN function
        noisy_received_Power = mean((noisy_received_Signal)^2)
        features_Powers[instance] = watts2dbm(noisy_received_Power)
```

Algorithm 1 SimLocRF Dataset Generation in MATLAB - Part 2

```
2.1.2. Generate Features as Spiking Frequencies from Pre-Processing Stage
  [Power, spiking_Frequency] = loadPostLayoutData()
                                          ▷ Load Data from pre-processing Stage
  features_Frequencies = interp1(Power, spiking_Frequency,
features_Powers)                                        ▷ Interpolate Features
2.2. Generate labels
  labels = zeros(parameters.num_angles *
parameters.num_instances_per_angle, parameters.num_labels)
        ▷ Assigns two labels: angle and distance (distances omitted here for
simplicity, values discussed in thesis = 0.1, 0.3, 0.5 m)
  for i = 1 to parameters.num_angles
    for n = 1 to parameters.num_instances_per_angle
2.2.1. Identify Region Index and Angle Index within Region
      region_index = floor(theta_s[i] / parameters.region_range)
        ▷ Range of angles per region given by: parameters.region_range =
parameters.num_angles / parameters.num_regions
      angle_within_region = rem(theta_s[i] ,
parameters.region_range)
      angle_within_region_index = floor(angle_within_region /
parameters.resolution)
      if (region_index >= parameters.num_regions) then
        region_index = 0                    ▷ Adjust for angle wrap-around
2.2.2. Assign Label
      labels[i * n, :]  = [region_index, angle_within_region_index]
```