Code 1:

**Simulation Environment Setup:**

The code starts by defining the simulation environment, which includes the following parameters:

enWidth, enLength, and enHeight: These parameters specify the width, length, and height of the simulation area.

maxHeighUAV and minHeighUAV: These parameters set the maximum and minimum allowed heights for the UAV.

noUsers, noBS, and noUAV: These parameters determine the number of ground users, base stations, and UAVs in the simulation.

**Ground Users, Base Stations, and UAV Positions:**

The code initializes the positions of ground users, base stations, and UAVs. Specifically, it sets the initial and final positions for the UAVs, initial positions for UAV movement, and positions for base stations. Additionally, random positions are generated for ground users within the simulation area.

**Path Loss Model and Rician Fading Model Information:**

The code defines parameters for the path loss model and the Rician fading model. These parameters include:

eta: The path loss component.

b\_0dB and b\_0: The reference channel gain in dB and linear scale, respectively.

k: An additional attenuation factor for non-line-of-sight (NLoS) conditions.

K\_min and K\_max: Parameters for the Rician fading model.

A1 and A2: Rician fading model parameters.

g: A complex Gaussian random variable used for Rician fading.

**Simulation:**

LoS (Line of Sight) and NLoS (Non-Line of Sight) channel characteristics between UAVs and base stations (BSs) and between UAVs and ground users. It accounts for path loss, elevation angles, and channel gains.

Power coefficients for ground users based on channel conditions.

Achievable rates for ground users in both predefined UAV trajectories and optimized UAV trajectories.

The optimization of UAV trajectories: The code checks for minimum achievable rates among all ground users and adjusts the UAV's horizontal position to improve communication performance.

Visualization of UAV trajectories and minimum achievable rates.

**Conclusion:**

The code generates various plots to visualize the achievable rates for different ground users in predefined and optimized UAV trajectories. It shows how the achievable rates change over time as the UAV moves.

Code2:

This code appears to optimize the trajectory based on different objectives: energy efficiency (EE-max), rate maximization (Rate-max), and energy minimization (Energy-min).

**Parameters Setup:**

You define various parameters such as the initial and final positions (q0 and qF), initial and final velocities (v0 and vF), maximum and minimum velocities (Vmax and Vmin), maximum acceleration (amax), total time (T), time step (deltat), and gravitational acceleration (g). These parameters are crucial for defining the problem.

**Communication Parameters:**

You also set up parameters related to communication, including bandwidth (B), noise power spectral density (N0dBm), noise power (sigma2), transmit power (PdBm), and channel gain (beta0dB).

**Trajectory Optimization:**

The code then calls three different trajectory optimization functions: EE\_A1, RM\_A1, and EM\_A1. These functions are used to optimize the trajectory based on the specified objectives and constraints. Each function likely implements a different optimization algorithm.

**Trajectory Plotting:**

After obtaining the optimized trajectories for the three objectives, the code plots the UAV's trajectory in a figure labeled "Trajectory." It uses the plot function to display the trajectories in different styles (solid line, dash-dot, and dashed) and assigns legends and labels to the plot.

**Speed Plotting:**

The code also plots the speed of the UAV over time in a figure labeled "Speed." It calculates the speed using the norms function and plots the speed profiles for the three different trajectory optimization objectives.

**Visualization and Styling:**

The code uses various MATLAB functions to add labels, titles, legends, and gridlines to the plots for better visualization and understanding.

**EE\_A1.m:**

Objective: This code is focused on energy-efficient trajectory optimization for a UAV.

Approach:The initial trajectory is set as straight-line flight with constant speed, and the code aims to optimize this trajectory for energy efficiency.It uses Sequential Convex Approximation (SCA) and Quadratic Transform (QT) methods for optimization.

It iteratively refines the trajectory to maximize energy efficiency while satisfying velocity, acceleration, and communication constraints.

Key Functions:QT.m: Implements the Quadratic Transform method to solve fractional programming for trajectory optimization.

Output:Optimized trajectory (q), velocity (v), and acceleration (a).

Average rate (Raver), average power consumption (Paver), and energy efficiency (EE).

**RM\_A1.m:**

Objective: This code is focused on maximizing the data rate of a UAV's trajectory.

Approach: It also uses SCA and Quadratic Transform methods for optimization.

The initial trajectory (q\_EE) is provided, and the code refines it to maximize data rate while respecting constraints on velocity, acceleration, and communication quality.

Key Functions: QT.m: Similar to the EE\_A1 code, it uses Quadratic Transform for trajectory optimization.

Output:Optimized trajectory (q), velocity (v), and acceleration (a).

Average rate (Raver), average power consumption (Paver), and energy efficiency (EE).

**EM\_A1.m:**

Objective: This code is focused on minimizing energy consumption for a UAV's trajectory.

Approach: Like the other codes, it employs SCA for trajectory optimization.

The initial trajectory (q\_EE) is used as a starting point, and the code iteratively improves it to minimize energy consumption, taking into account velocity, acceleration, and communication constraints.

Key Functions:No additional key functions are used.

Output:Optimized trajectory (q), velocity (v), and acceleration (a).

Average rate (Raver), average power consumption (Paver), and energy efficiency (EE).