ChatGPT

ISAC仿真系统总体架构

设计一个集成感知与通信（ISAC）的 MATLAB 仿真系统，需使用 Phased Array System Toolbox、Communications Toolbox 和 Signal Processing Toolbox 等工具箱[1]。系统主脚本为 `ISAC_Main.m`，其余功能模块分别存放于独立 `.m` 文件中。主要模块包括：天线阵列与波束形成、OFDM 通信链路、FMCW 感知波形与目标建模、目标跟踪、频谱动态分配（基于贝叶斯强化学习）等。系统参数（如阵列元素数、时隙步数、用户数、可视化开关等）在主脚本中统一配置，并可灵活调整。

- **天线阵列与波束形成**：使用 `phased.ULA` 等系统对象创建 64 元阵列（或可配置的元素数）[2]。采用 `phased.SteeringVector` 或 `phased.PhaseShiftBeamformer` 等对象计算波束权重，实现基于指定角度的波束指向控制[3][4]。例如，主脚本中可调用模块函数 `ComputeBeamWeights.m`，输入阵列对象与目标指向角，输出权重向量。

- **OFDM 通信链路**：采用 Communications Toolbox 中的 OFDM 模块实现调制解调。可使用 `comm.OFDMModulator` / `comm.OFDMDemodulator` 系统对象，或直接调用 `ofdmmod` / `ofdmdemod` 函数[5]。支持参数可配置：子载波数、循环前缀长度、调制阶数（QAM）等。将生成的 QAM 符号映射到子载波，例如使用 `qammod`（取代已弃用的 `comm.RectangularQAMModulator`[6]）进行星座图调制。接收端使用相应的解调器和信道估计恢复数据，并通过 `comm.ErrorRate` 对象计算误比特率（BER）[7]。信噪比（SNR）可通过 MATLAB 的信号处理函数 `snr` 或直接根据接收信号和噪声功率比值计算得到[8]。模块可命名为 `OFDM_Link.m`，输出各用户的 BER、SNR 及通信速率等指标。

- **FMCW 感知波形与目标建模**：采用频调连续波（FMCW）雷达波形，用于目标探测和测距。通过 `phased.FMCWWaveform` 对象生成可调参数的 FMCW 信号[9]。构建目标模型可使用 `phased.RadarTarget` 对象，根据目标雷达截面积（RCS）和 Swerling 型号模拟目标反射[10]。在仿真中，将发射信号通过目标模型生成回波，再添加噪声和多径散射。模块可称为 `FMCW_Sensing.m`，实现感知侧波形生成和信号回波模拟。检测概率 P_d 可通过 CFAR 探测后统计命中率得到（对比真目标数与检测数，或基于 SNR 阈值公式估算），依赖模拟结果计算。

- **目标跟踪**：对感知到的目标进行跟踪，估计目标位置和速度。可采用 MATLAB 的跟踪滤波器，如 `trackingKF`（线性卡尔曼滤波器）或 `trackingEKF`（扩展卡尔曼滤波器）对象[11]。比如在模块 `TargetTracking.m` 中，维护目标状态（位置、速度），根据测量更新滤波器，实现目标轨迹跟踪。用户可配置目标初始角度、距离和运动模型。

- **频谱动态分配（贝叶斯强化学习）**：将频谱占用比例 η 视为状态/动作决策变量，使用强化学习算法动态调整资源分配。MATLAB Reinforcement Learning Toolbox 提供了多种 RL 算法（如 DQN、PPO 等）可用于资源分配优化[12]。在本系统中，可自行实现基于贝叶斯更新的 RL 引擎，维护对环境状态（如当前用户负载、干扰情况等）的概率分布，通过试验-更新机制优化 η。例如，设计模块 `SpectrumRL.m`，输入当前系统速率、检测性能等信息，输出下一步的 η 值；可结合贝叶斯方法更新策略以应对不确定性。此处重点是架构，可引用 RL 工具箱文档说明支持复杂控制策略[12]。

- **多用户与环境配置**：系统允许自定义用户数量及其运动参数。每个用户具有独立的角度、距离、速度或运动轨迹参数，可在主脚本中通过输入变量配置。通信链路模块按照用户数生成多路 OFDM 信号，并对每个用户采用不同波束指向。目标跟踪模块可根据用户及散射点参数模拟多目标场景。

- **性能指标与数据保存**：仿真过程统计并输出关键性能指标，包括总吞吐率、各用户速率、通信误码率（BER）、感知检测概率 P_d、频谱占用率 η 等。主脚本可将这些结果保存在结构体或数组中，

最后使用 `save('Results.mat','metrics')` 将变量保存为 `.mat` 文件供后续分析（MATLAB 的 `save` 函数用于将工作区变量存储到 MAT 文件）。

- **可视化**：构建独立模块 `Visualization.m` 用于绘图展示结果。可视化内容包括：阵列波束方向图（使用 `phased.ULA.patternAzimuth` 或 `patternAzimuth` 方法绘制二维波束图 [13]）、用户和目标运动轨迹（3D 轨迹图）、频谱分配随时间的变化曲线、BER 随时隙演化图等。波束图例可调用 `patternAzimuth(array,Fc,el)` 绘制零度仰角处的方向图 [13]。轨迹可用 `plot3` 或 `scatter3` 等函数绘制。确保在主脚本中根据配置参数（如 `enableVisualization` 开关）调用可视化模块。

- **模块化实现**：项目采用模块化结构。主脚本 `ISAC_Main.m` 负责参数初始化和各模块调用流程，例如：

```
% ISAC_Main.m (示例结构)
% 1. 参数设置
params.Narray = 64;
params.numUsers = 3;
params.subcarriers = 1024;
params.modOrder = 16; % QAM阶数
params.numSteps = 100;
params.enableVis = true;
% 2. 初始化阵列对象
params.array = phased.ULA(params.Narray, λ/2);
% 3. 仿真循环
for t = 1:params.numSteps
  % (a) 通信：生成数据并 OFDM 调制 -> 发射
  [txSignal, dataBits] = OFDM_Link(txConfig, params, users(t));
  % (b) 发射通过信道加噪并接收 -> 解调 -> BER, SNR 计算
  [rxData, metricsComm] = OFDM_Receive(rxConfig, txSignal);
  % (c) 感知：发射 FMCW 波形并接收回波 -> 检测目标
  [returnSignal] = FMCW_Sensing(fmcwConfig, targetModels);
  metricsSense = Radar_Process(returnSignal);
  % (d) 目标跟踪更新
  trackedTargets = TargetTracking(metricsSense, prevTargets);
  % (e) 频谱决策：根据当前性能更新 η
  η = SpectrumRL(metricsComm, metricsSense, prevState);
  % (f) 记录指标
  results(t).BER = metricsComm.BER;
  results(t).SNR = metricsComm.SNR;
  results(t).Pd = metricsSense.Pd;
  results(t).eta = η;
  % 可视化
  if params.enableVis
    Visualization(params, beamWeights, trackedTargets, η, metricsComm);
  end
end
% 4. 保存结果
save('ISAC_results.mat','results');
```

上述代码仅为示例框架。实际实现时，各部分细节对应具体函数，例如 `OFDM_Link.m` 内实现 OFDM 调制并返回发送信号，`OFDM_Receive.m` 内进行通道模型和解调，`FMCW_Sensing.m` 内利用

`phased.FMCWWaveform` 和 `phased.RadarTarget` 生成回波，`Radar_Process.m` 进行距离-多普勒处理并计算 P_d 等，`SpectrumRL.m` 执行强化学习决策。

- **注意事项**：所有 MATLAB 代码基于 R2023a 版本编写。由于新版工具箱弃用了一些旧函数，例如 `comm.RectangularQAMModulator`，应使用新函数 `qammod` 进行 QAM 调制 [6]。保证代码兼容性和无警告。

核心模块说明与参考

- **天线与波束权重计算**：使用 `phased.ULA` 创建阵列对象，并通过 `phased.SteeringVector` 计算导向矢量，再对其求共轭得到权重 [3]。也可直接使用 `phased.PhaseShiftBeamformer` 来形成指定角度的波束 [4]。例如，`sv = phased.SteeringVector('SensorArray',arrayObj,'PropagationSpeed',c); weights = conj(sv(fc,[az;el]));` 可得到指向指定方位角的波束权重。

- **OFDM 调制解调**：使用 Communications Toolbox 的 `ofdmmod` 和 `ofdmdemod` 或 `comm.OFDMModulator` 对象生成 OFDM 信号 [5]。例如：`ofdmMod = comm.OFDMModulator('FFTLength',Nfft,'NumGuardBandCarriers',[Ng1 Ng2],...); txSig = ofdmMod(qamData);`。接收端使用对应的 `comm.OFDMDemodulator` 进行解调。编码时，用 `qammod(bits, M, 'InputType','bit','UnitAveragePower',true)` 产生星座点 [14]。误码率通过 `comm.ErrorRate` 统计 [7]，SNR 可用 `snr(rxSignal, noise)` 计算 [8]。

- **FMCW 波形与目标**：采用 `phased.FMCWWaveform` 生成基带 FMWCW 信号 [9]，可配置扫频带宽、扫频时间、采样率等。目标采用 `phased.RadarTarget` 模型，其平均雷达截面积、模型（Swerling 型）可设置 [10]。信号传播与接收可用 `phased.FreeSpace` 或自定义信道模型模块进行，叠加高斯噪声后形成接收回波。

- **目标跟踪**：利用 MATLAB 的跟踪滤波器，如 `trackingKF` 对象对目标进行位置、速度估计 [11]。如设定常速模型为 2D或3D 常速度，通过测量更新状态估计，从而实现对动态目标的连续跟踪。

- **可视化**：参考 `phased.ULA.patternAzimuth` 函数绘制阵列方向图 [13]。例如：`patternAzimuth(arrayObj,fc,0,'Azimuth',-180:180)` 绘制零仰角下方位方向图。目标轨迹和用户位置可用 MATLAB 的 `plot3`、`quiver` 等函数绘制。通信性能如 BER 变化曲线可通过 `plot(1:T, BER_vector)` 展示时序演变。

参考文献

- MathWorks 官方示例和文档，介绍了基于 MIMO-OFDM 的通信中心 ISAC 系统模型 [1]、OFDM 调制函数 [5]、误码率计算 [7]、FMCW 波形生成 [9] 等方法。
- MATLAB Answers 和帮助文档指出最新版本用 `qammod` 替代已弃用的 `comm.RectangularQAMModulator` [6]。
- Phased Array Toolbox 文档介绍了 `phased.SteeringVector`、`phased.PhaseShiftBeamformer` 等对象如何计算阵列波束 [3] [4]；`phased.ULA.patternAzimuth` 提供了阵列方向图绘制方法 [13]。
- 跟踪滤波器帮助文档说明了 `trackingKF` 等对象可用于目标运动状态估计 [11]。
- Reinforcement Learning Toolbox 文档指出其可用于训练资源分配策略 [12]，为频谱控制策略实现提供基础支持。

上述资料和示例为实现所需模块提供了参考和基础，具体代码逻辑可根据系统需求在 MATLAB 2023a 中逐步构建、测试和验证。

---

1 2 14 Integrated Sensing and Communication II: Communication-Centric Approach Using MIMO-OFDM - MATLAB & Simulink
https://www.mathworks.com/help/phased/ug/integrated-sensing-and-communication-2-communication-centric-approach-using-mimo-ofdm.html

3 phased.SteeringVector - Sensor array steering vector - MATLAB
https://www.mathworks.com/help/phased/ref/phased.steeringvector-system-object.html

4 phased.PhaseShiftBeamformer - Narrowband phase shift beamformer - MATLAB
https://www.mathworks.com/help/phased/ref/phased.phaseshiftbeamformer-system-object.html

5 OFDM Modulation Using MATLAB - MATLAB & Simulink
https://www.mathworks.com/help/comm/gs/ofdm-modulation-using-matlab.html

6 Why do I receive "Error using RectangularQAMModulator" - MATLAB Answers - MATLAB Central
https://www.mathworks.com/matlabcentral/answers/1885517-why-do-i-receive-error-using-rectangularqammodulator

7 comm.ErrorRate
https://www.mathworks.com/help//releases/R2021a/comm/ref/comm.errorrate-system-object.html

8 snr - Signal-to-noise ratio - MATLAB
https://www.mathworks.com/help/signal/ref/snr.html

9 FMCW Waveforms - MATLAB & Simulink
https://www.mathworks.com/help/phased/ug/fmcw-waveform.html

10 phased.RadarTarget - Radar target - MATLAB
https://www.mathworks.com/help/phased/ref/phased.radartarget-system-object.html

11 trackingKF - Linear Kalman filter for object tracking - MATLAB
https://www.mathworks.com/help/radar/ref/trackingkf.html

12 Reinforcement Learning Toolbox
https://www.mathworks.com/help//releases/R2021a/reinforcement-learning/index.html

13 phased.ULA.patternAzimuth - Plot ULA array directivity or pattern versus azimuth - MATLAB
https://www.mathworks.com/help/phased/ref/phased.ula.patternazimuth.html