# SEMTA Radar Simulation System

## Overview

### Summary

* This document describes the use and functionality of the SEMTA simulation, signal processing, post-processing, and testing system implemented in MATLAB.

### Requirements

* General:
  + MATLAB
  + Signal Processing Toolbox
  + Image Processing Toolbox
    - Minor dependency, used for connected component analysis algorithm in Functions/Detection/Detection.m
* Radar Simulation:
  + Communications Systems Toolbox
  + Phased Array Systems Toolbox
* Fixed Point Implementation (Required for modification but not execution):
  + Fixed Point Designer
  + DSP Toolbox
* Parallel Processing:
  + Parallel Computing Toolbox

## User Guide:

### Setup

### Execution

## File & Function Description

### File Directories:

* **Definitions** – Contains the definition of the “RadarScenario” class
  + *Radar Scenario.m* - This class is used as the main scenario object which holds all relevant simulation data, and contains methods which mostly perform plotting & visualization
* **Documentation** – User-written documentation
  + *Radar Simulation.docx* – This document
  + *Post Processing Server.docx* – Describes the post-processing Python server
* **Figures** – Output folder for *SaveFigures* function. Not included in git repo.
* **Fixed Point –** Contains files used by Fixed Point Designer tool
  + **codegen**  - Files generated by the FP Designer, including fixed point versions of MATLAB functions, wrappers to use these functions in floating point code, and .mex binary files which are compiled with fixed point data types.
  + **Function** – Functions written as input to the FP Designer tool
  + **Project** – Project files, which open FP designer and document conversion process
  + **Reference** – .mat files containing test bench inputs and reference values
  + **Test Bench** – Scripts which test the functions in “Function” and “codegen”
* **Functions –** Functions used in main system
  + **Data Processing –** Functions which operate on single-unit data in coordinate form, after signal processing.
    - *BeamSteeringUpdate.m –* Updates beamsteering based on tracking results and current steering mode
    - *KalmanFilter\_SingleUnit.*m – Kalman filter for single-unit tracking
    - *MahanalobisDistance.*m – Statistical distance measurement used in fine gating for single-unit tracking
    - *ModeCheck.*m – Updates beamsteering mode using tracking results
    - *Tracking\_SingleUnit.*m – Outer function for single unit tracking, including Kalman filter, fine and course gating, and track management.
    - *Tracking\_SingleUnit\_Bidirectional.*m – Single-unit tracking which is capable of processing both forward in time and backwards in time. Used for post-processing.
    - *Tracking\_SingleUnit\_Post.*m – Runs both forward and backwards tracking, and performs data fusion. Used for post-processing.
  + **Multistatic Processing –** Functions which operate on data derived from multiple radar units, after single-unit Data Processing
    - *DataFusion.m –* Uses inverse-variance-weighted averaging to combine results from single-unit tracking systems.
    - *Tracking\_Multi.*m – Outer function for multiple unit tracking, running both forwards and backwards in time and implementing a Kalman filter.
    - *Tracking\_Multi\_Bidirectional.*m – Runs both forwards and backwards tracking, and performs data fusion.
  + **Signal Processing –** Functions which operate on raw received signals or the radar data cube.
    - *Detection.m –* Performs CFAR detection and data integration on radar power cube, then performs coordinate estimation using a centroid algorithm and an amplitude monopulse algorithm.
    - *SignalProcessing.*m – Range and doppler processing of input signals
    - *SignalProcessing\_FixedPoint.*m – Same as previous function but with fixed-point range and doppler processing functions implemented
    - *SimulateAxes.*m – Generates range and doppler axes without running signal processing. Used for scenarios without simulation.
  + **Simulation –** Functions used in simulation of the real radar scenario, not used in a real test setup
    - *CalculateSNR.m –* Calculate ideal SNR of point target at given coordinates. Used to verify signal responses of simulation system.
    - *CalculateVariance.m –* Using empirical curves from Monte Carlo tests, estimate error variance of measurement at set coordinates. Used to initialize variance matrices in Kalman filter.
    - *PhasedSetup.*m – Sets up Phased Array Toolbox objects used by simulation
    - *RadarSimulation*.m – Runs simulation of single frame radar transmission, reflection, and reception. This is the main script for simulating received signals.
    - *SimulateDetections*.m – Using empirical results from simulation tests, generates detection coordinates without running simulation. Used for rapid Monte Carlo testing.
    - *TargetRCSModel.m* – Implements fluctuating target RCS model with specular reflection.
    - *TrajectoryModel.m* – Implements trajectory simulation for target motion.
  + **Utilities –** Useful functions such as file handling and memory management
    - *EmailAlert.m* – Sends email from external SMTP server. Useful for notifying user if long tests are complete.
    - *EmailSetup*.m – Sets up email alert system.
    - *SaveFigures*.m – Saves open figures to file.
    - *SaveScenario*.m – Saves scenario data object to file
    - *SaveTracking*.m – Saves tracking results to file, for use in post processing Python server.
* **PostProcessing –** External post-processing HTTP server running in Python. See *Documentation/Post Processing Server.docx* for more information.
  + **Input –** Files uploaded to HTTP server. Not included in git repo.
  + **Output –** Files generated by post processing server. Not included in git repo.
  + *Server.py* – HTTP server which collects files and initializes post processing.
  + *Tracking*.py – Post processing and result generation functions.
  + *requirements*.txt – Python package list for pip.
* **References –** Reference documents
  + **Phase I** – Documents from SBIR Phase I
    - *SEMTA Kickoff\_190930.pdf –* Slides from Phase I kickoff
    - *SEMTA Signal Processing.*docx – Phase I signal processing description
  + *33265 AF-SEMTA II Kickoff Meeting 210629.pdf –* Slides from Phase II kickoff
* **Results –** Folder used to save results by *SaveScenario* and *SaveTracking.* Not included in git repo.
* **Scratch** – Assorted scratch files. Not maintained.
* **Scripts** – Matlab programs, not written as functions, used for setup and execution of system.
  + **Functional** – Starting tasks, stopping tasks, and main simulation loop
    - *Main.m* – Main simulation/processing loop
    - *Main\_FixedPoint*.m – Same as above, but calls fixed point signal processing system
    - *Main\_NoSimulation*.m – Same as above, but generates detection coordinates form empirical curves instead of running simulation.
    - *StartProcess.m –* Tasks to run at beginning of simulation/processing.
    - *EndProcess.*m – Tasks to run at end of simulation/processing .
  + **Setup** – Scripts which declare parameters and simulation setup
    - *SetupRadarScenario.m –* Declares parameters and variables of radar system, waveform, processing, and tracking
    - *SetupTarget.*m – Declares RCS and trajectory of target.
    - *SetupMulti.*m – Declares number of receivers, their positions, and the number of frames.
    - *SetupSimulation.*m – Sets up parameters of system, such as whether to use parallel processing or whether to save files.
* **Test** – Files used for Monte Carlo testing. Not maintained.
* *FullSystem.m –* Main script for running system.
* *FullSystem\_FixedPoint.*m – Main script for running system with integrated fixed point functions
* *FullSystem\_NoSimulation.*m – Main script for running system, skipping simulation and signal processing.

## Notes

### Open Tasks