# SEMTA Simulation & Processing Description

## Overview

The simulation and processing system is intended to perform end-to-end simulation of all aspects of the SEMTA system. This includes physical simulation of the radar signal response from the target, signal processing of the radar data cube, data processing of radar measurements, and multistatic data processing of the results of multiple radar units. Block diagram overview of the system is shown in Figure 1.

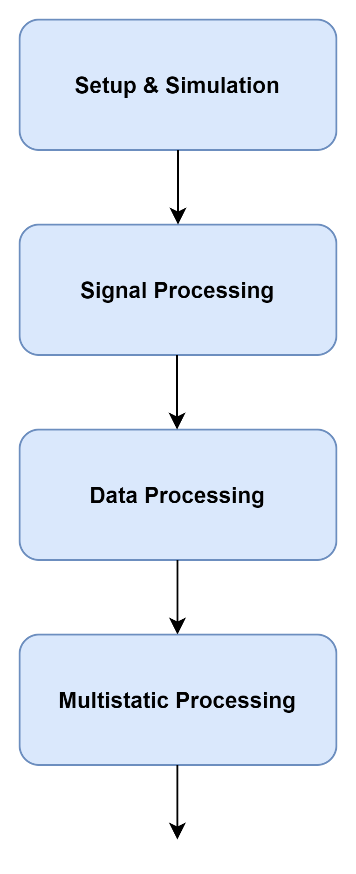


Figure 1. Basic overview of system operation.

## Radar Signal Response Simulation

The first segment of the SEMTA system is the simulation component. This segment is intended to provide receive signal data which can then be used as input to the signal processing system, validating its correct operation. The simulation system is intended to be self-contained, so that it can be replaced with a real data parsing system without requiring large changes in the code base.

The position of the target and of each radar unit is determined by a few parameters, illustrated in Figure 3. The target’s trajectory is defined by a speed along the “track”, the distance of its sinusoidal excursion from the center of the track, and the period of the excursion. The array of radar units is defined by the distance from each unit to the center of the target’s trajectory, and by the distance between each radar unit.

The target uses a fluctuating RCS model which simulates the dependence of a real target’s RCS on the frequency and aspect angle of the reflection. The purely fluctuating component of the RCS characteristic is generated by calculating the RCS of an ensemble of point scatterers which occupy the same cross-sectional area as the real target. In addition, a peak caused by specular reflection of a semi-cylindrical aircraft is added to the RCS. Plot of an example target RCS is shown in Figure 4.

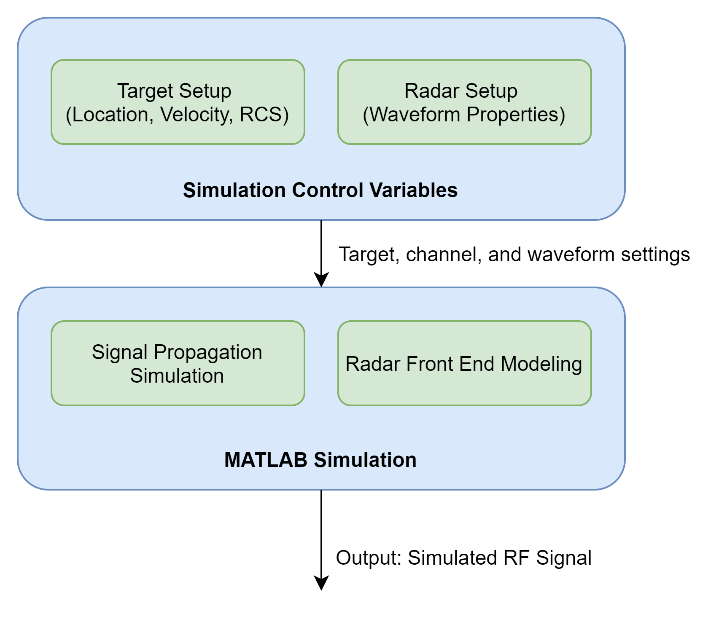


Figure 2. Simulation system components.

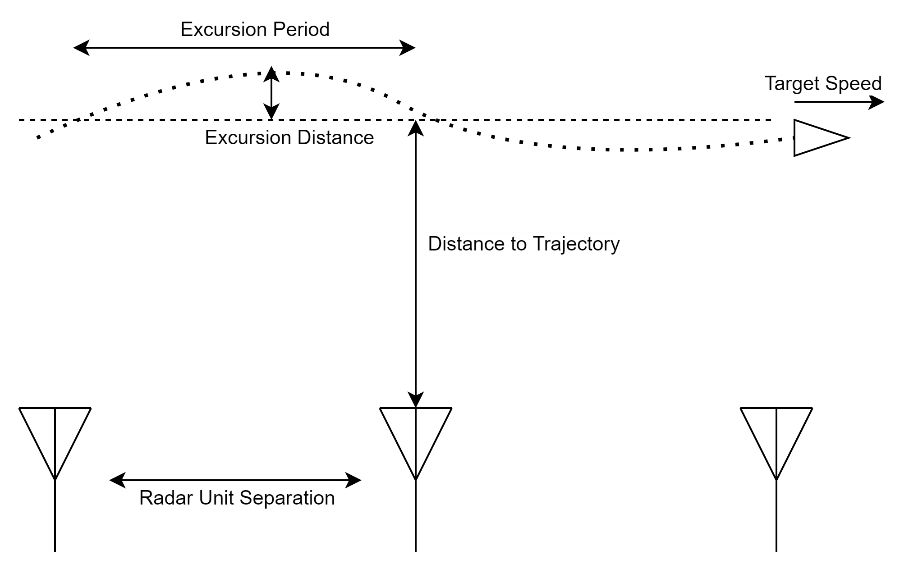


Figure 3. Target and radar system positioning.

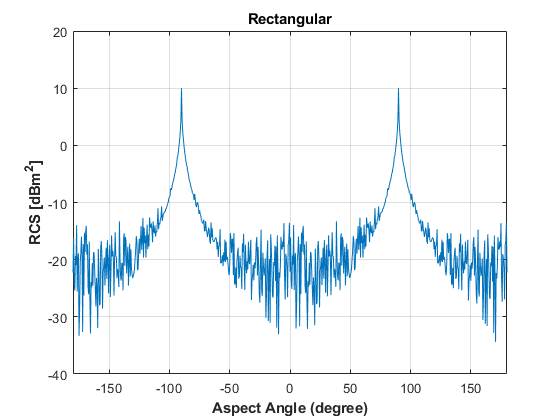


Figure 4. Example target RCS model.

The MATLAB simulation system allows the waveform and timing parameters of the radar system to be modified as needed. Nominal values of the waveform parameters are shown in Table 1 below. The SEMTA system uses a linear frequency modulated waveform, pulsed at a set pulse repetition frequency, across each coherent processing interval.

|  |  |
| --- | --- |
| Parameter | Nominal Value |
| Center Frequency | 9.45GHz |
| LFM Bandwidth | 10MHz |
| Pulse Duration | 10µs |
| Pulse Repetition Frequency | 20kHz |
| Pulses Per CPI | 1024 |
| CPI Per Frame (Search Mode) | 5 |
| CPI Per Frame (Track Mode) | 1 |

Table 1. Nominal waveform parameters used for this investigation.

Simulation of radar transmission, reflection, and reception is implemented using tools from the Mathworks “Phased Array Toolbox” for MATLAB. This library provides functions which simulate antenna patterns, channel effects, signal reflections, and receiver characteristics. Using these tools, a receive signal is calculated from the transmit waveform, and sent to the input of the signal processing system.

## Signal Processing

The signal processing system performs operations on the radar data cube during each CPI of operation. This includes range processing, Doppler processing, CFAR detection, integration, and coordinate estimation. Block diagram of this segment is shown in Figure 5. While the previous system is used to generate simulated input data, this system replicates the functionality of the radar unit’s FPGA signal processing system.

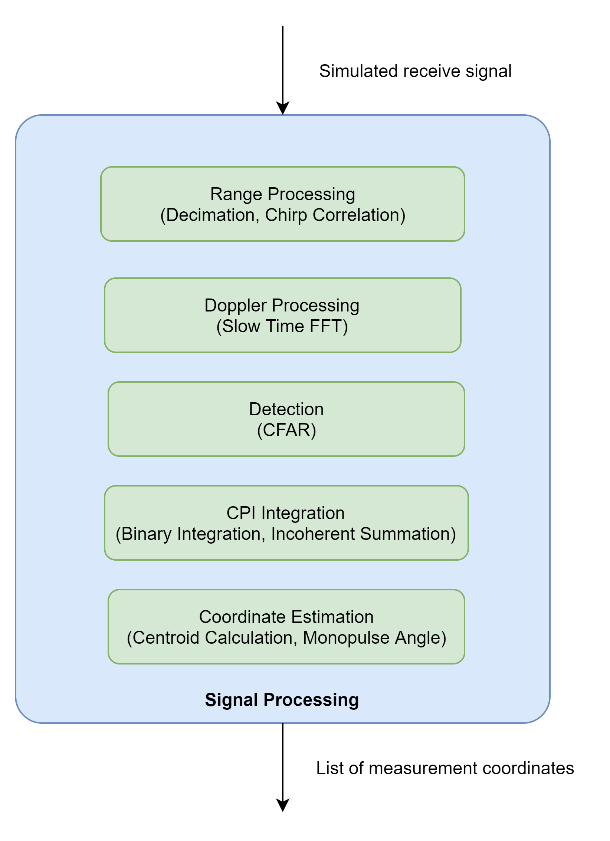


Figure 5. Overview of signal processing system.

Range processing is applied to the simulated receive signal, by performing a correlation with a windowed version of the transmit signal. Doppler processing is performed by taking the FFT of the range processed data in the slow time dimension, resulting in the range-Doppler data cube. Detection is performed using a 2D-CFAR algorithm.

The system must perform both binary m-of-n integration and incoherent integration, depending on whether the radar unit is in search or track mode. In the case of binary integration, a range bin migration compensation algorithm is applied to all detections, to mitigate the effects of high-speed targets moving between range bins between CPI. This is done by extending each detection through the swath of range bins that a target in that bin would cover by moving at the given Doppler velocity for the remainder of the frame. An illustration of this compensation algorithm is shown in Figure 6.

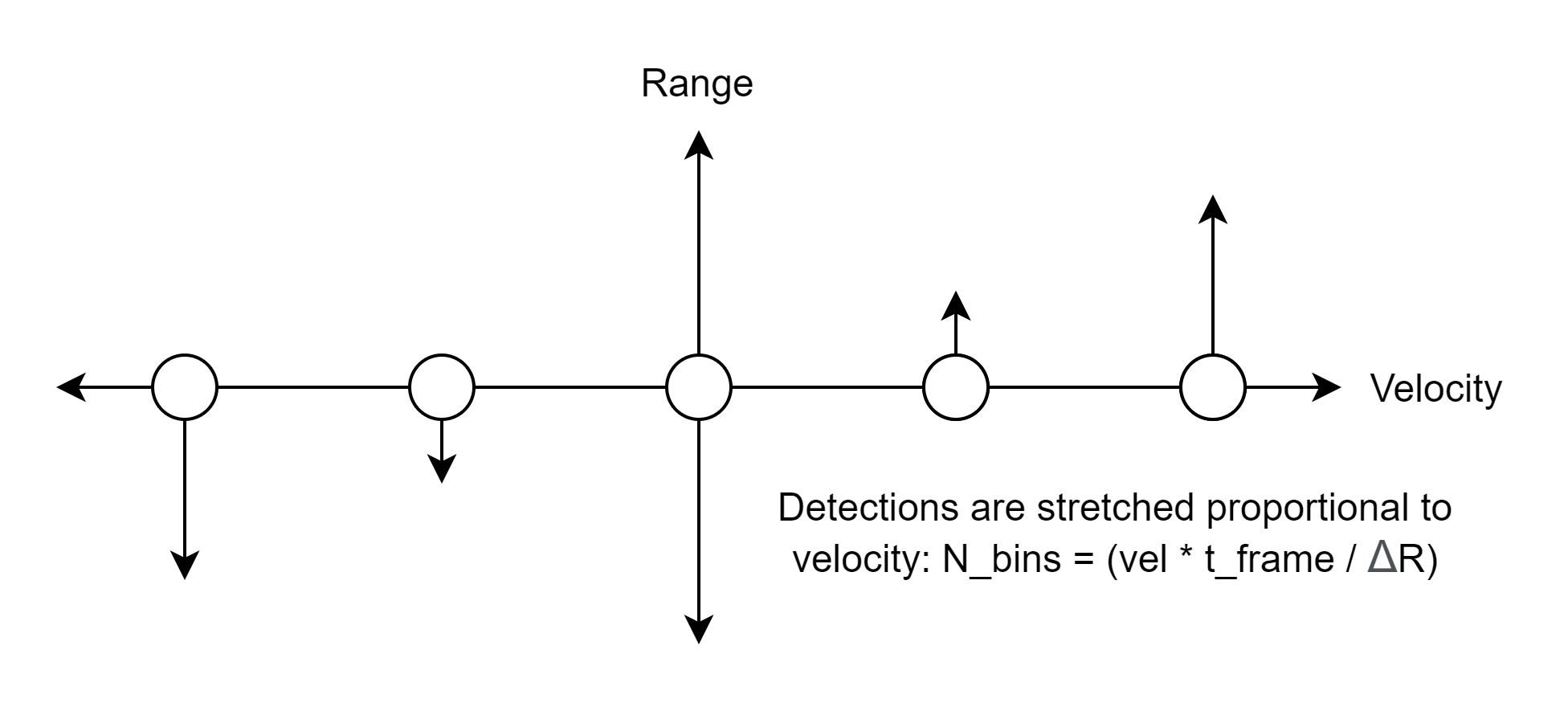


Figure 6. Illustration of range bin migration compensation used in binary integration algorithm.

After integration is performed, the “Hoshen-Kopelman algorithm” is used to determine all connected components of the binary detection data cube. Each connected component is treated as a single detection, which is passed to the coordinate estimation algorithm.

Coordinate estimation is performed using a power-weighted centroid method. The centroid value is computed for the target’s range and velocity by using its position in the data cube, and an estimation of the target’s angle-of-arrival is computed by taking a power-weighted average of the amplitude monopulse value in each bin of detection.

The range, velocity, and angle estimations are saved to a list of targets along with a timestamp and signal power estimate. This list of targets is then provided as the input to the data processing system.

## Single Unit Data Processing

Using the list of estimated target coordinates, each unit performs gating and track association, followed by tracking using a Kalman filter. Block diagram overview of data processing system is shown in Figure 7.

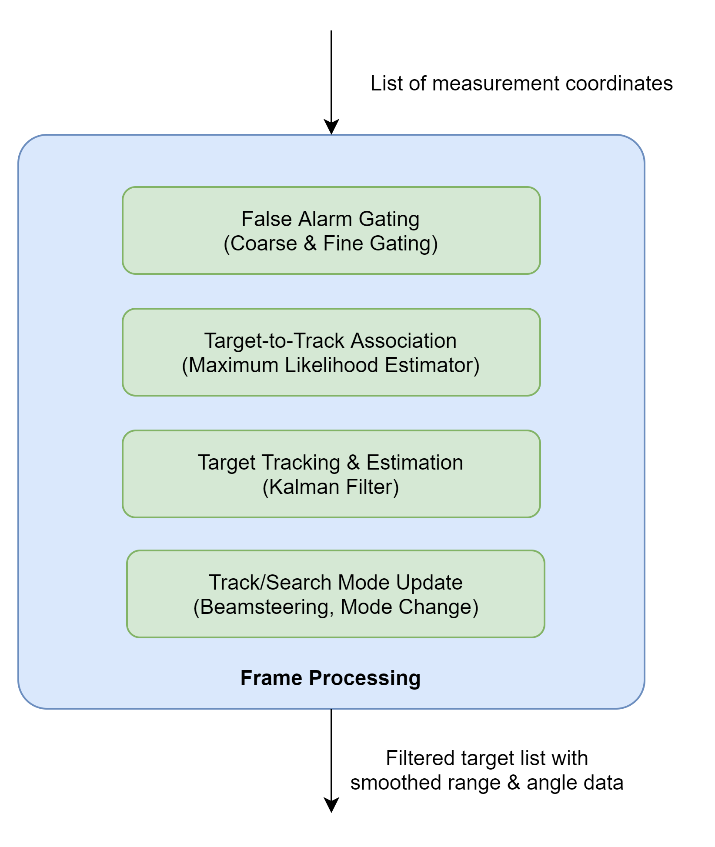


Figure 7. Block diagram of single unit data processing system.

Both coarse and fine gating to determine if any detections are not plausible as new inputs to existing tracks. Coarse gating eliminates all detections which have a larger change in range than is possible during the frame time, based on a maximum target velocity parameter. Fine gating takes the “Mahanalobis distance”, a measure of the statistical distance between a measurement and a distribution, using the previous track’s covariance matrix.

Target-to-track association is performed by using the remaining measurement which has the lowest Mahanalobis distance from the predicted location of the tracked target. This measurement is used as input to the Kalman tracking filter, along with an estimate of the error uncertainty of the measurement which is calculated using empirical curves determined by previous Monte Carlo testing.

In order to model the frame-to-frame behavior of the radar system, there is a single callback from the results of the tracking system. The estimated angle of the next target position is used to set the simulated beamsteering angle, and whether the tracking system counts a new detection or a missed detection is used to determine whether the system should simulate the radar in track mode or in search mode.

### Multistatic Data Processing

Block diagram overview of the SEMTA post-processing system is shown in Figure 8 below. The first step in the post-processing system is performing tracking on the results of each individual radar unit. During live processing, this is done during each frame of operation, sending new radar measurements as input to the live tracking system, which performs coarse and fine gating, target-to-track association, and Kalman filter tracking.

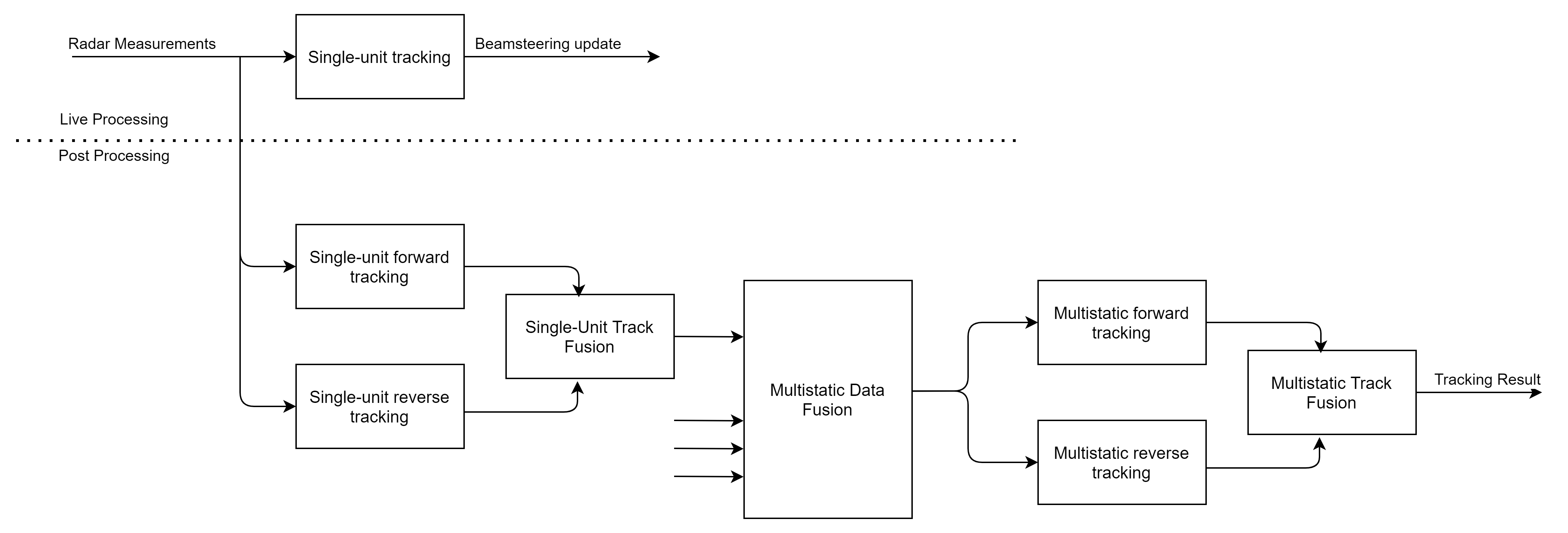


Figure 8. Block diagram overview of post-processing system

For post-processing, the system has the benefit of performing smoothing as well as tracking by applying Kalman filter estimation both forward in time and backwards in time. Additionally, since gating and target-to-track association has been performed by the single unit system, these functions can be bypassed, using only the radar measurements which passed track association.

The Kalman filter used is a two-dimensional, second-order Kalman filter. This means that the filter maintains a state vector which contains the position, velocity, and acceleration of the target in both x- and y-directions. Along with the state vector, a covariance matrix is maintained which measures the uncertainty of the tracked estimate. This uncertainty is used to calculate the Kalman gain, which interpolates between the prediction and the measurement at each frame.

Since the Kalman filter uses both a state measurement and its covariance, the raw radar measurements (R, θ, and vd) and variances (σR, σθ, σvd) must be converted to cartesian form. For an existing track, only the position measurement is required, which can be calculated directly as follows:

However, when a new track is initialized, its full state and covariance must be initialized with a full estimate of position, velocity, and acceleration state and covariance. This can not be done entirely analytically, as the doppler velocity only measures radial motion relative to the radar unit, and there is no measurement of acceleration. To accommodate this, the velocity normal to the direction of the radar is treated as a uniform random variable, **V**n, as is the acceleration in either direction, **A**.

Each uniform random variable is treated as symmetric, with limits ±vmax and ±amax provided as set parameters to the tracking system. For a symmetric uniform distribution, the mean and variance can be calculated as follows.

Using these assumptions, the initial velocity and acceleration can be set as follows:

For each radar unit’s data, the forward and reverse Kalman filters are applied, and the two results are fused using inverse-variance weighted averaging, which produces both an improved state estimate and a reduced variance estimate, which is used to seed the multistatic tracking system.

To combine the results of multiple radar systems, all smoothed measurements are combined into a vector of measurement data objects, which is then sorted in order of the measurement’s time stamp. A second pass of tracking is then performed, applying both a forward and reverse tracking system to the combined vector of measurements, and again using inverse-variance weighted averaging to produce the final result.