Radar Tracking Waveform Design in Continuous Space and Optimization Selection Using Differential Evolution

by

Bryan Paul

A Thesis Presented in Partial Fulfillment
of the Requirement for the Degree
Master of Science

Approved April 2014 by the
Graduate Supervisory Committee:

Antonia Papandreou-Suppappola, Daniel W. Bliss, Chair
Cihan Tepedelenlioglu

ARIZONA STATE UNIVERSITY

May 2014

ABSTRACT

Waveform design that allows for a wide variety of frequency-modulation (FM) has proven benefits. However, dictionary based optimization is limited and gradient search methods are often intractable. A new method is proposed using differential evolution to design waveforms with instantaneous frequencies (IFs) with cubic FM functions whose coefficients are constrained to the surface of the three dimensional unit sphere. Cubic IF functions subsume well-known IF functions such as linear, quadratic monomial, and cubic monomial IF functions. In addition, all nonlinear IF functions sufficiently approximated by a third order Taylor series over the unit time sequence can be represented in this space. Analog methods for generating polynomial IF waveforms are well established allowing for practical implementation in real world systems. By sufficiently constraining the search space to these waveforms of interest, alternative optimization methods such as differential evolution can be used to optimize tracking performance in a variety of radar environments. While simplified tracking models and finite waveform dictionaries have information theoretic results, continuous waveform design in high SNR, narrowband, cluttered environments is explored.

To Francis

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support of many people. Many thanks to my advisers, Dr. Antonia Papandreou-Suppappola and Dr. Daniel Bliss, as well as the rest of my committee, Dr. Cihan Tepedelenlioglu. Thanks to Brian O'Donnell and John Kota with their help with the model. Thanks to my co-workers who put up with me rambling on about this theory or that conjecture. My previous employer, Validus Technologies, for funding the first half of my degree and being flexible with classes, tests, and other obligations. My current employer, General Dynamics Advanced Information Systems, for funding the second half of my degree and being equally flexible and understanding. Thanks to my parents David Paul and Betsy Paul, who always support me in every endeavor. Thank you to my sister Megan Knedler, brother Matthew Paul and the rest of my family who keep me going. Thanks to many of the close friends who provide much needed respite throughout my journey. Finally, thanks to my loving wife Aimee Paul, who without her support this thesis would not even be close to a reality. You are my shining light!

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Chapter 1

INTRODUCTION

Tracking in cluttered environments is challenging [2]; measurements that are processed could be from a false alarm due to noise in the radar receiver, cluttered returns, or the target. Attempts to minimize tracking error in challenging environments has been explored extensively [3–9]. While simplified scenarios such as clutter-free, one-dimensional tracking applied information theoretic results [10], closed form solutions are often not possible in realistic environments such as two or three-dimensional tracking in cluttered environments [4], where nonlinear relationships exist between the state space and the measurement model [11]. Further, these solutions employed finite waveform libraries containing only linear chirps. Nonlinear frequency-modulated (FM) signals have been explored in finite libraries in more complicated models [3], but these too contained a limited number of fixed shape waveforms. In addition, traditional gradient-based optimization methods proved mathematically intractable, and so grid-based approaches were used to explore the time-bandwidth space. While this was a great improvement over previous works, the library was limited to grid points chosen in advance, and so potentially valuable waveforms between the grid points could never be chosen. Alternative methods of optimization for waveform design, such as genetic algorithms, have been explored to reduce integrated sidelobe levels and improve the ambiguous free range of radar systems [12]. However, they were employed only over the set of bi-phase and poly-phase fixed frequency and fixed amplitude waveforms. More general waveform spaces for generating nonlinear waveforms have been investigated in other works [13], but the effort has been restricted to fixed waveforms for application-specific radar that do not vary with time as the tracking dynamics change.

To build on previous works in adaptive radar, two key elements are required: a parameterized waveform that can take on the shape of most waveforms of value for radar tracking, and an efficient method of optimization to minimize the predicted mean-squared error (MSE) (PMSE) in this new parameter space. Clearly, these two elements are interrelated. A waveform must be parameterized such that it is conducive to efficient optimization. In addition, the optimal waveform can vary drastically with respect to the environment, and so the optimization process must allow for a rich set of waveforms to be available for selection. In this thesis, we develop a parameterizable cubic polynomial FM (CP-FM) function and demonstrate how it can be constrained for efficient optimization using the genetic algorithm differential evolution (DE) to generate nonlinear FM signals in a continuous space to minimize the PMSE in complex, cluttered tracking scenarios.

The thesis is organized as follows. In Chapter 2, we provide an overview of nonlinear target tracking using particle filtering, and we discuss waveform design to improve tracking performance under adverse environmental conditions. In Chapter 3, we introduce the spherically-constrained cubic frequency-modulated waveform, and we propose the use of the genetic algorithm differential evolution to select the waveform's optimal parameters in Chapter 4. Our application of selecting the waveform parameters at each time step by minimizing the predicted mean-squared error of tracking estimation is demonstrated in Chapter 5.

Chapter 2

SYSTEM MODEL

We want to estimate the physical state of a slowly fluctuating point target with a linear motion model in two-dimensional (2D) space using a discrete-time constant velocity model [14]:

$$\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \\ y_{k+1} \\ \dot{y}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \end{bmatrix} + q_k \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 & 0 \\ \frac{\Delta t^2}{2} & \Delta t & 0 & 0 \\ 0 & 0 & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ 0 & 0 & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix} \tag{2.1}$$

where x_{k+1} is the target position along the x-axis at discrete time step $k+1$, y_{k+1} is the target position along the y-axis at discrete time step $k+1$, $\dot{x}_{k+1}$ is the target velocity projected on the x-axis at discrete time step $k+1$, $\dot{y}_{k+1}$ is the target velocity projected on the y-axis at discrete time step $k+1$, Δt is the discrete time step duration of the system (duration between time steps k and $k+1$), $\begin{bmatrix} x_k & \dot{x}_k & y_k & \dot{y}_k \end{bmatrix}^T$ is the state vector at discrete time k, and q_k is the process model error intensity at discrete time k.

Note that although we assume the velocity is constant (and therefore acceleration is zero), in reality the small perturbations of the velocity modeled by the dynamical error covariance matrix allow for small variations in acceleration. This is the stochastic state estimation approach that assumes, by construction, that the motion model contains some error [2]. In this sense, changing velocity is tracked assuming there is uncertainty in the motion model. The recursive Bayesian network therefore includes this probability cloud in likelihood decisions, allowing for a measurement that

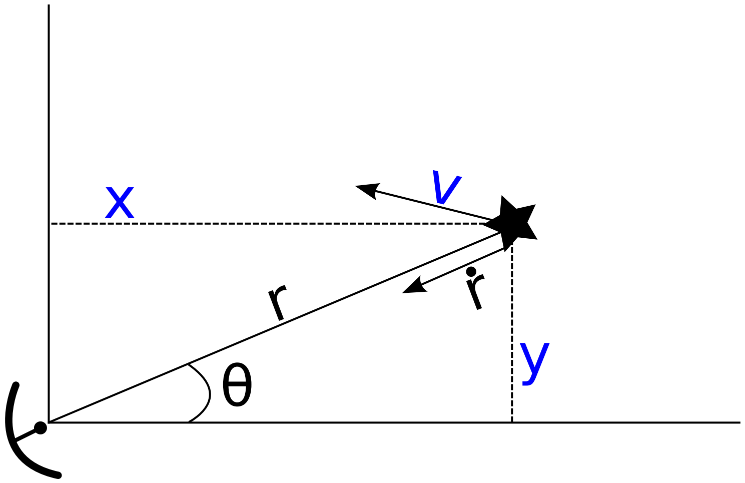

Figure 2.1: Monostatic, 2D radar geometry

indicates a change in velocity to drive probability away from the constant velocity track in response to changes in actual velocity. Though this simplifies the model (and subsequently the tracking process), it means that large deviations in velocity may be difficult to handle. Nevertheless, the particle filter (PF) we plan on using allows for more sophisticated models [11, 14] to be included in the framework, and so results derived here can be extended by estimating the unknown changes in velocity. For tracks that exhibit significant changing velocity, the process noise intensity q_k can be increased. This results in higher error during stationary (non-changing velocity) portions of the track. However, they allow for the Bayesian processor to explore with higher probability deviations from the constant velocity track that may actually occur with the target to prevent divergence. This also helps with the initial acquisition of the target since the precise mean of the target is not known at the initial time instance. Since the system time step is Δt, we drop the discrete qualifier when discussing time in the future.

We assume a monostatic 2D radar configuration as shown in Figure 2.1. The radar is at the origin in this case, and the target location in the (x, y) plane is marked by the star. The state parameters to be estimated are the position in 2D space (x, y), and the velocity, which has an x and y component. The observed parameters, after Doppler processing [2], are the range (r_k), range-rate $(\dot{r}_k)$, and bearing (θ). If the sensor is not at the origin, all calculations can be done with respect to the origin by subtracting the sensor location from the respective coordinates. We assume a narrowband environment such that only a frequency shift (Doppler shift) is induced in the returned waveform, not the more general Doppler time scaling. This is possible under the assumption [15]:

$$TB \ll \frac{c}{2\dot{r}_k} \tag{2.2}$$

where c is the speed of electromagnetic waves in air, T the signal duration, and B the signal bandwidth. This means that the range and range-rate which offset the peak of the narrowband cross-ambiguity function at the measurement processor relate to the position and velocity state as follows [15, 16]:

$$r_k = \sqrt{x_k^2 + y_k^2} \tag{2.3}$$

$$\dot{r}_k = \frac{x_k \dot{x}_k + y_k \dot{y}_k}{\sqrt{x_k^2 + y_k^2}} \tag{2.4}$$

$$\theta_k = \arctan\left(\frac{y_k}{x_k}\right) \tag{2.5}$$

We assume a high signal-to-noise ratio (SNR) environment, where the maximum likelihood estimator (MLE) of the range and range-rate come from the location of the peak of the narrowband ambiguity function, which attains the Cramér-Rao lower bound (CRLB) [15]. As a result, the covariance of the estimate depends on the main lobe shape of the narrowband ambiguity function, and thus the transmitted waveform [15]. This variance is calculated using the inverse of the Fisher information matrix (FIM) [8]:

$$\mathbf{R}_{vv} = \frac{1}{\text{SNR}} \mathbf{T}\mathbf{U}^{-1}\mathbf{T} \tag{2.6}$$

for $\mathbf{T} = \text{diag}(c/2, c/(4\pi f_c))$ and the elements of $\mathbf{U}$ defined as:

$$\begin{aligned} u_{11} &= \textstyle\int_{-\infty}^{\infty} \omega^2 \left|\tilde{S}(\omega)^2\right| d\omega/2\pi \\ u_{12} &= \textstyle\int_{-\infty}^{\infty} t\phi'(t) \left|\tilde{s}(t)^2\right| dt \\ u_{21} &= \textstyle\int_{-\infty}^{\infty} t\phi'(t) \left|\tilde{s}(t)^2\right| dt \\ u_{22} &= \textstyle\int_{-\infty}^{\infty} t^2 \left|\tilde{s}(t)^2\right| dt \end{aligned} \tag{2.7}$$

$\tilde{s}(t)$ is the envelope of the pulse, $\tilde{S}(\omega)$ is the Fourier transform of the envelope of the pulse, and $\phi(t)$ is the phase function of the pulse. Note that the bearing angle measurement is assumed to be independent of the range and range-rate obtained in the Doppler processing step. This is because we can assume that it was obtained from a separate measurement that is not a part of the returned waveform, such as an electro-optical (EO) sensor [7].

The final predicted mean-squared error (MSE) (PMSE), which serves as the cost function for the optimization process, is a function of this measurement error (for a given waveform), the previous state estimate, the predicted state estimate, the predicted measurement estimate, the clutter density, and the predicted SNR. After each time step, we estimate the PMSE for the parameters we wish to test in the differential evolution (DE) optimization step. The test parameters with the lowest cost value are used in the next time step. Note that the CRLB shown in Equation 2.6 is only achieved using the estimator asymptotically as the number of samples $N \to \infty$, not for SNR approaching infinity [15]. As a result, we assume that many pulses must be sent to achieve this bound, a common requirement for extracting the Doppler shift for realistic range-rate values in narrowband environments [2]. More accurate global bounds can be found to more accurately portray the single pulse radar system, but at great additional computational cost, since the full ambiguity function must be characterized [4, 15].

We employ a form of data fusion to incorporate information about the target dynamics (physical motion limits), as well as the measurement. This is because the measurement is corrupted by noise, and the motion model may not be perfect. For example, the assumption that the target has a constant velocity may not be true, and in this case, incorporating and heavily weighting the measurement can be vital to maintaining track. As a converse, the measurement may be particularly noisy, or obscured by cluttered detects. In this case, if the motion model accurately reflects the target dynamics, incorporating and heavily weighting the predicted target state can be vital to maintaining track. We use the bootstrap particle filter (BFP) to recursively propagate the Bayesian belief of the target state and ultimately sample from the filtering distribution [11]. The algorithm is outlined in Table 2.1. Here we take the initial state, which is perturbed by Gaussian noise with covariance $\mathbf{R}_{rr}$, and create N_p particles of this state. We perturb each one of these by the same covariance to represent statistical trials as to where the true state could be. This set of particles completely characterizes the statistics of the initial state, if we have enough of them to draw accurate inferences from them using Monte Carlo methods [11]. In the prediction step, we propagate these trials through the motion model perturbing them by the model covariance. This allows for the statistical spread of belief based on the previous state, the physics of the target, and the uncertainty of the physics between measurements. We then take the predicted particles and pass them through the actual measurement model. The result is a set of trials (particles) that exhibit the statistics of the measurement. Once a measurement is received, the likelihood function is used to weigh these predicted measurements based on how statistically likely the measurements could have originated from the particle's state. This is complicated by the presence of multiple detections from the target and the clutter, or just clutter. From these weights, we can infer the true state by performing a nor-

Initialization	
$\underline{\hat{x}}[0\|0]$ (*a priori* knowledge)	State
$\hat{\mathbf{P}}[0\|0]$ (*a priori* knowledge)	Covariance
$\underline{\chi}_i[0\|0] = \underline{\hat{x}}[0\|0] + \underline{n}_i,\ \underline{n}_i \sim \mathcal{N}(\underline{0}, \mathbf{R}_{rr})$	Particles
$W_i = \frac{1}{Np}$	Particle Weights
Prediction	
$\underline{\chi}_i[n\|n-1] = \mathbf{A}\underline{\chi}_i[n-1\|n-1] + \underline{r}_i,\ \underline{r}_i \sim \mathcal{N}(\underline{0}, \mathbf{R}_{ww})$	Particle State
$\underline{\Upsilon}_i[n\|n-1] = \mathbf{c}\left[\underline{\chi}_i[n\|n-1]\right]$	Particle Measurement
Update	
$W_i = \frac{1}{\sqrt{(2\pi)^3 \|\mathbf{R}_{vv}\|}} e^{\frac{1}{2}\left(\underline{y}[n]-\underline{\Upsilon}_i[n\|n-1]\right)^T \mathbf{R}_{vv}^{-1}\left(\underline{y}[n]-\underline{\Upsilon}_i[n\|n-1]\right)}$	Particle Weights
Inference	
$\underline{\hat{x}}[n\|n] = \frac{\sum_{i=1}^{N_p} W_i \underline{\chi}_i[n\|n-1]}{\sum_{i=1}^{N_p} W_i}$	MMSE State Estimate
Resampling	
$\underline{\chi}_i[n\|n-1] \leftarrow \underline{\tilde{\chi}}_i[n\|n-1]$	Particle Distribution

Table 2.1: The bootstrap particle filter algorithm

malized, weighted sum of the predicted particles. That is, the stronger the likelihood a predicted measurement from a particular particle, the more heavily weighted that particular particle is. Note we could also do other means of inference, such as the mode of the distribution by picking the particle with the largest weight. Finally, to avoid particle degeneracy [11], we resample at every time step. This is easily accomplished using inverse transform sampling [11], essentially using the existing particle set to drive the statistics of the next set. In particular, heavily weighted particles are repeated (sampled) more than low weighted particles.

We use a Poisson process as is typically assumed for probabilistic data association [17] to model the potential clutter interferers. Since we have a motion model and ideally a target governed by the laws of physics, we can create a region around the predicted measurement mean by which the measured target should fall within with high probability. This is called the validation gate [17], and is used to filter clutter while retaining the target with high probability [18]. Since we are concerned with clutter returns only in the target measurement space, that is the returned range and range-rate, the validation gate region is a 2D ellipsoid with the major and minor axis derived from the measurement variance of the range and range-rate sensors [3]. Rotation of the ellipse is governed by the correlation coefficients, or the off-diagonal elements of the measurement covariance matrix. Since we wish to enclose the measurement with high probability, we choose a standard deviation multiplier of $\gamma = 4$ [18]. Then the validation gate volume is calculated as [19]:

$$V_g = \pi \left|\gamma^2 \mathbf{S}_k\right|^{1/2} \tag{2.8}$$

where $\mathbf{S}_k$ is the predicted measurement error covariance added to the measurement covariance (obtained from the FIM and scaled by the predicted SNR).

To simulate this volumetric clutter, we must first randomly decide how many clutterers are enclosed in the volume. This is done by drawing from a Poisson process:

$$m_k \sim \text{Pois}\left(\rho V_g\right) \tag{2.9}$$

where ρ is the clutter density. Then, once we know how many clutterers are in the validation gate, we must uniformly sample m_k points from the 2D ellipse. For previous works [3], an approximation was used:

$$\begin{aligned} p_x &\sim U(-\sqrt{u_1\gamma^2}, \sqrt{u_1\gamma^2}) \\ p_y &\sim U(-\sqrt{u_2(\gamma^2 - \tfrac{p_x^2}{u_1})}, \sqrt{u_2(\gamma^2 - \tfrac{p_x^2}{u_1}})) \end{aligned} \tag{2.10}$$

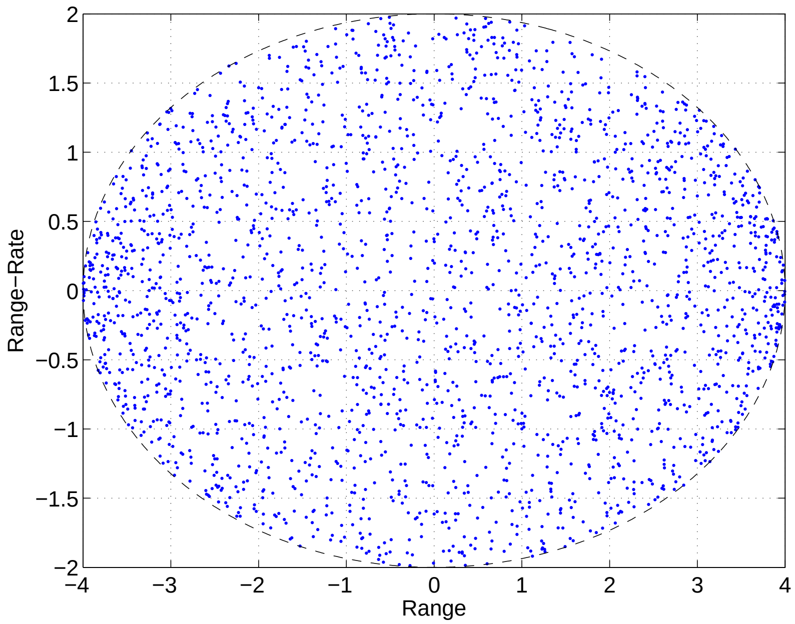

Figure 2.2: Example of sampling in an ellipse, previous methods

where u_1 and u_2 are the eigenvalues of the predicted measurement error covariance matrix, $\mathbf{S}_k$ [3].

An example of the sampling is shown in Figure 2.2. Note that the rotation of the ellipse is obtained using the eigenvalues of $\mathbf{S}_k$ and applied to the clutter afterwards [3]. Unfortunately, this is biased, as shown in the histogram in Figure 2.3. This is a common problem with ellipse transformed uniform data, in that the transformation results in a non-uniform distribution over the new ellipsoid space [20]. For the range/range-rate case, measurements appear to be biased toward the end points of the first axis generated using that technique. As a result, clutter tends to group closer to the edges of the validation volume. This potentially has multiple problems. First we note that clutter is more likely to be centered on the predicted range-rate measurement, so for constant velocity tracks, clutter contributes constructively with the averaged estimate when using probabilistic data association (PDA), which uses a statistically weighted sum to compute the final soft state estimate. In addition, range clutterers are at the edges of the validation gate region, far from the predicted mea-

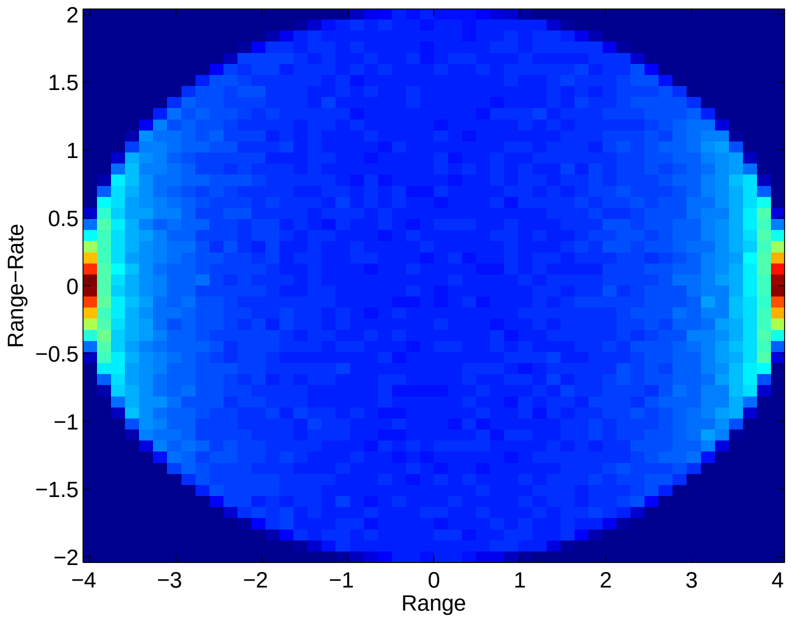

Figure 2.3: Histogram of sampling in an ellipse, previous methods

surement at the center of the ellipse. They are therefore weighted much lower than the potential measurement (if detected and contained within the validation gate), and contribute less to the weighted statistical average. Finally, pairs clutterers on opposite ends of the range ellipse cancel out in the average due to their opposing signs and near equal probability. As a result, the clutter is under-modeled from the true uniform volumetric distribution.

We instead use rejection sampling [20] by generating samples from a rectangle that has the ellipse inscribed in it that we wish to sample from, and discarding samples that fall outside of the ellipse. This allows for uniform sampling inside the ellipse, but we pay an efficiency penalty of:

$$\frac{A_{\text{rectangle}}}{A_{\text{ellipse}}} = \frac{2a2b}{\pi ab} = \frac{4}{\pi} \tag{2.11}$$

This means that on average, we need to generate ${}^{4}/_{\pi}$ as many samples to ensure we obtain m_k samples inside of the ellipse. Since this is an average, and it is relatively

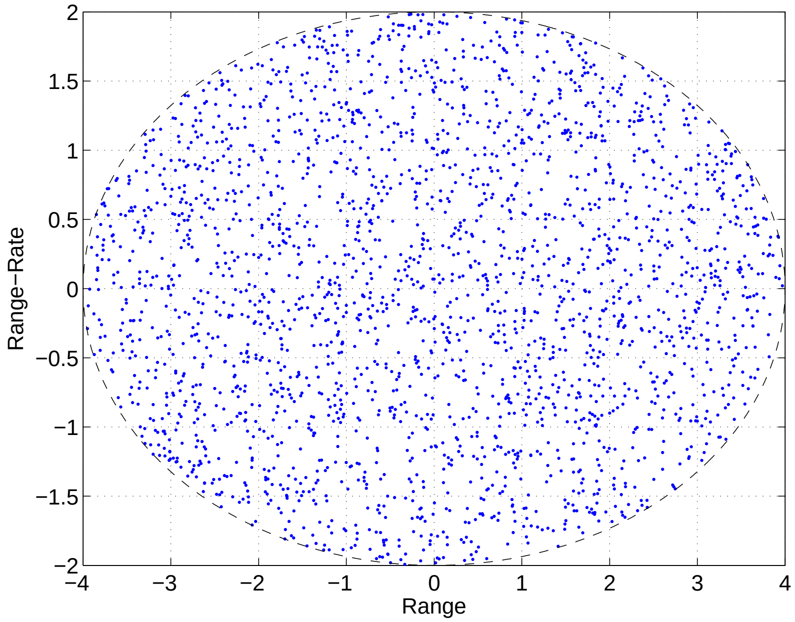

Figure 2.4: Example of rejection sampling an ellipse

low cost to generate these points, we generate twice this amount and then verify we have enough points. If not, we generate again, although this is very unlikely. In addition it is worth emphasizing that we are merely attempting to model an unwanted interference, and so this particular processing element would not be in the active design. For an ellipse with semi-major and semi-minor axis of a and b respectively representing the x and y axis (without loss of generality), the rejection criteria is:

$$\text{Reject} : \frac{p_x^2}{a^2} + \frac{p_y^2}{b^2} - 1 > 0 \tag{2.12}$$

To estimate the predicted error at each time step, clutter needs to be incorporated into the model. Unfortunately, this requires knowledge of the number of false alarms from clutter and their location, something not known before the return of the waveform [18]. As a result, the spreading covariance used in the PDA filter [17] can only be approximated on average. Even these approximations involve, in general, complex integrals. Some have calculated these integrals using Monte Carlo methods [9], es-

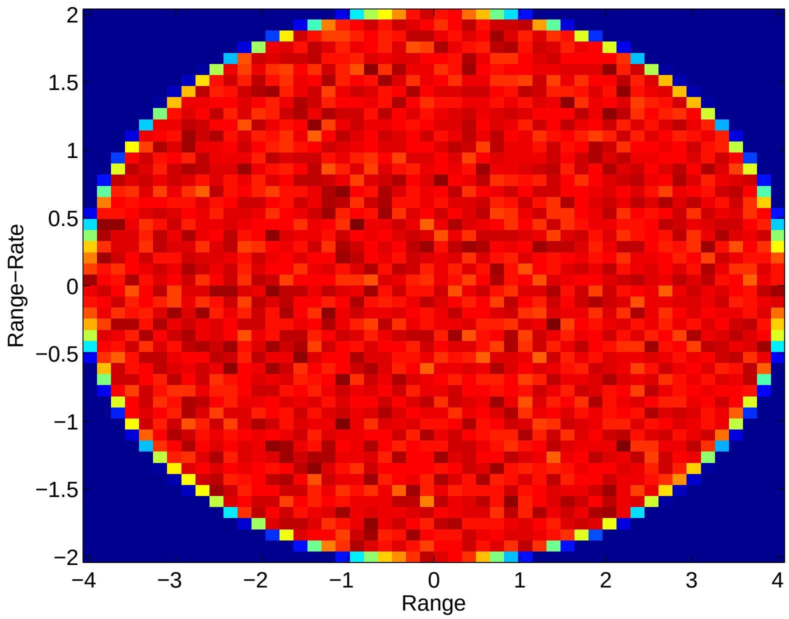

Figure 2.5: Histogram of rejection sampling an ellipse

sentially employing a PF. However, this can be shown to be, with some simplifying assumptions [18]:

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - q_2 \mathbf{K}\mathbf{R}_{vv}\mathbf{K}' \tag{2.13}$$

The factor q_2 still contains a potentially infinite summation to capture the probability of potentially infinite clutterers in the validation gate. Others [3] have therefore used an approximation to this factor [1] based on truncating the sum:

$$q_2 = \frac{0.997P_D}{1 + 0.37P_D^{-1.57}\rho V} \tag{2.14}$$

However, as noted in the article itself, this approximation may be poor at high detection probabilities, something we expect given the high SNR assumption. The article also computes, using a truncated summation of the infinite sum in Equation 2.13, tabular values for the information reduction factor. Since we are already approximating this covariance, and given the high SNR assumption, a better method of computing is sought out. Therefore, we make the decision to use the table in [1] and

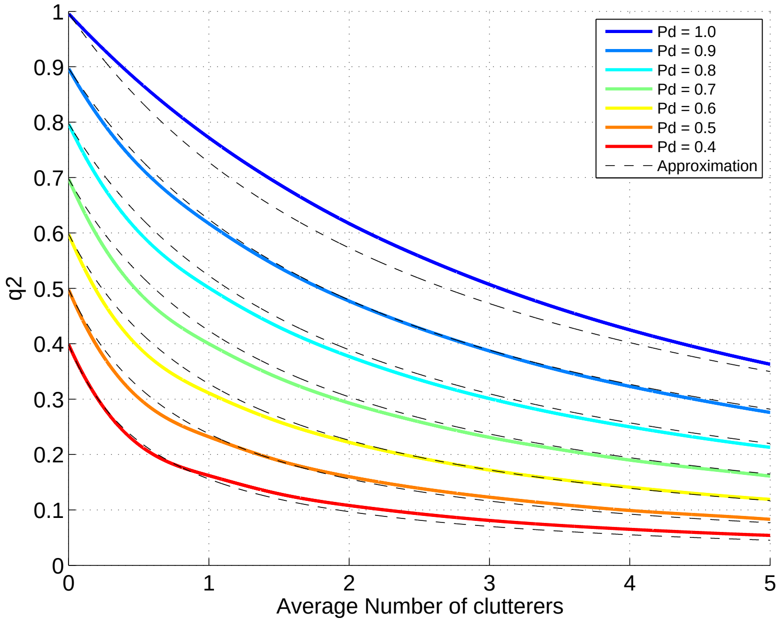

Figure 2.6: Comparison of interpolation vs formulaic calculation of q_2

perform cubic spline interpolation and extrapolation to compute the clutter contribution to the reduction in the information. In doing so, we were able to recreate the plot in [1] as shown in Figure 2.6. Here, the colored lines are interpolated from the table via cubic splines, while the dashed lines represent the formulaic approximation proposed by [1] for the closest colored line probability of detection. The table used in the interpolation is recreated here for reference in Table 2.2.

As we can see from Equation 2.13, this method still requires estimation of $P_{k|k-1}$ and R_{xy} (for the Kalman gain K), and so we use the unscented transform (UT) [21]. Though we could potentially reuse the prediction steps in the standard particle filter to be used in this time step, this allows for a decoupling between the waveform design and the tracking process, since one could use other tracking methods instead of the particle filter. To help prevent numerical breakdown during the Cholesky decomposition required for the matrix square root, we use the scaled unscented Kalman filter (UKF) [11] given in Table 2.3. Note that $\kappa = (3\alpha^2 - 4)$ under the Gaussian prior assumption and for a state space of dimension 4 [11]. A good set of parameters

	Probability of Detection						
ρV	**1.0**	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**	**0.4**
0.0	0.399	0.498	0.598	0.698	0.797	0.897	0.996
0.5	0.218	0.302	0.394	0.493	0.602	0.722	0.873
1.0	0.162	0.232	0.311	0.400	0.501	0.617	0.772
1.5	0.129	0.189	0.259	0.338	0.430	0.538	0.688
2.0	0.108	0.160	0.222	0.293	0.377	0.477	0.617
2.5	0.093	0.139	0.194	0.259	0.335	0.428	0.558
3.0	0.081	0.123	0.172	0.231	0.301	0.387	0.507
3.5	0.072	0.110	0.155	0.209	0.273	0.352	0.463
4.0	0.065	0.099	0.141	0.190	0.250	0.323	0.425
4.5	0.059	0.091	0.129	0.175	0.230	0.298	0.392
5.0	0.054	0.083	0.119	0.161	0.213	0.276	0.363

Table 2.2: Values of q_2 for M = 2, g = 4, from [1]

are $\alpha = 10^{-3}$ and $\beta = 2$, the latter of which is optimal for Gaussian distributions [11].

While the particle filter we use is the standard particle filter, the final likelihood is weighted against the non-detect, clutter only probability [22] using PDA [17]:

$$\tilde{W}_i = (1 - \hat{P}_D) P_{m_k} V_g^{-m_k} + \hat{P}_D P_{m_{k-1}} V_g \frac{1}{m_k} \sum_{i=1}^{N_D} \sum_{j=1}^{N_p} W_{i,j} \tag{2.15}$$

where $\hat{P}_D$ is the estimated probability of detection based on the range measurement, P_{m_k} is the probability of m_k clutterers (obtained from the probability mass function of the Poisson distribution), and $W_{i,j}$ represents the j^{th} particle weight from the i^{th} detection. Note that the first term encompasses the probability that either we did not detect (in the simulation, we remove the true detection from the set with probability $1 - P_D$), or the detection fell outside of the validation gate volume. The second term

Initialization	
$\hat{\underline{x}}[0\|0]$ (*a priori* knowledge)	State
$\hat{\mathbf{P}}[0\|0]$ (*a priori* knowledge)	Covariance
$\underline{\chi}_0[0\|0] = \hat{\underline{x}}[0\|0]$	
$\underline{\chi}_i[0\|0] = \hat{\underline{x}}[0\|0] + \left(\sqrt{(N_x+\kappa)\,\hat{\mathbf{P}}[0\|0]}\right)_i$	Σ Points
$\underline{\chi}_{i+N_x}[0\|0] = \hat{\underline{x}}[0\|0] - \left(\sqrt{(N_x+\kappa)\,\hat{\mathbf{P}}[0\|0]}\right)_i$	
$W_0(\mu) = \frac{\kappa}{N_x+\kappa}$	
$W_0(\sigma^2) = \frac{\kappa}{N_x+\kappa} + (1-\alpha^2+\beta)$	Σ Weights
$W_i = \frac{\kappa}{2(N_x+\kappa)}$	
Prediction	
$\underline{\chi}_i[n\|n-1] = \mathbf{A}\underline{\chi}_i[n-1\|n-1]$	Σ Points
$\hat{\underline{x}}[n\|n-1] = \sum_{i=0}^{2N_x} W_i(\mu)\underline{\chi}_i[n\|n-1]$	State
$\tilde{\underline{\chi}}_i[n\|n-1] = \underline{\chi}_i[n\|n-1] - \hat{\underline{x}}[n\|n-1]$	State Error
$\hat{\mathbf{P}}[n\|n-1] = \sum_{i=0}^{2N_x} W_i(\sigma^2)\tilde{\underline{\chi}}_i[n\|n-1]\tilde{\underline{\chi}}_i[n\|n-1]^T + \mathbf{R}_{ww}$	Covariance
$\underline{\Upsilon}_i[n\|n-1] = \mathbf{c}[\underline{\chi}_i[n\|n-1]]$	Measurement Σ Points
$\hat{\underline{y}}[n\|n-1] = \sum_{i=0}^{2N_x} W_i(\mu)\underline{\Upsilon}_i[n\|n-1]$	Measurement
$\underline{\xi}_i[n\|n-1] = \underline{\Upsilon}_i[n\|n-1] - \hat{\underline{y}}[n\|n-1]$	Residual
$\mathbf{R}_{\xi\xi}[n\|n-1] = \sum_{i=0}^{2N_x} W_i(\sigma^2)\underline{\xi}_i[n\|n-1]\underline{\xi}_i[n\|n-1]^T + \mathbf{R}_{vv}$	Residual Covariance
$\mathbf{R}_{\tilde{\chi}\xi}[n\|n-1] = \sum_{i=0}^{2N_x} W_i(\sigma^2)\tilde{\underline{\chi}}_i[n\|n-1]\underline{\xi}_i[n\|n-1]^T$	Cross-Covariance

Table 2.3: The unscented Kalman filter algorithm

is a summation of all the likelihoods, which is the basic PDA algorithm; we do not make a hard decision as to what the target is, but average over all detections weighted by their respective likelihoods.

Note that actual SNR, as well as the estimated SNR, are computed using the range and a reference range/SNR [8]:

$$\mathrm{SNR} = \frac{{r_{\mathrm{ref}}}^4}{r} \mathrm{SNR}_{\mathrm{ref}} \tag{2.16}$$

In this thesis we use the following parameters:

$$\begin{aligned}
\Delta t &= 0.25 \text{ s} \\
f_c &= 10400000000 \text{ Hz} \\
c &= 299792458 \text{ m/s} \\
P_{\mathrm{FA}} &= 0.01 \\
\mathrm{SNR}_{\mathrm{ref}} &= 1 \\
r_{\mathrm{ref}} &= 50000 \text{ m}
\end{aligned} \tag{2.17}$$

The parameters q_k and ρ vary with different test scenarios. Note that the estimated SNR is computed using the predicted range value [8]. The probability of detection, true and estimated, is calculated using [23]:

$$P_D = P_{\mathrm{FA}}^{\left(\frac{1}{1+\mathrm{SNR}}\right)} \tag{2.18}$$

While the P_D is in general degraded by correlation loss, we are assuming a perfect matched filter in time delay and Doppler shift since the ambiguity function peak is the estimator we use. Therefore, the filter mismatch loss is not modeled here. The transmitted waveform $s(t)$ is given as [15]

$$s(t) = \sqrt{2} A(t) Re \left[\sqrt{E_t} e^{j2\pi \left(\int\limits_{-T/2}^{T/2} f(t)dt + f_c t \right)} \right] \tag{2.19}$$

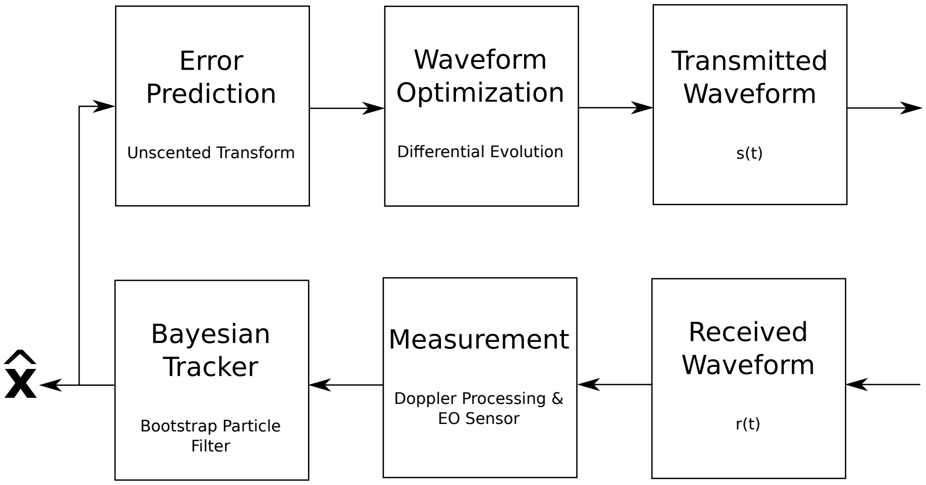

Figure 2.7: Top level system concept

while the received waveform is given as [15]

$$r(t) = \sqrt{2}A(t)Re\left[\sqrt{E_r}e^{j2\pi\left(\int\limits_{-T/2}^{T/2} f(t-\tau)dt+f_c(t-\tau)\right)}e^{j2\pi f_D}\right] \tag{2.20}$$

where $A(t)$ is a trapezoidal window as used in [3] with a $10^{-8}s$ rise and fall time. The complete system concept is shown in Figure 2.7. The initial state estimate $\hat{\mathbf{x}}_0$ is known *a priori* during the previous (unmodeled) process of detection and track creation. The covariance of this initial detection and estimation process, as well as the target dynamics (motion model) are used to create a predicted error using the unscented transform. From this, the DE waveform optimization step computes the measurement covariance for a series of candidate waveforms and uses this, in addition to the estimated SNR and predictive clutter error modeling, to select the candidate waveform. This waveform is then transmitted, and subsequently received, another step that is abstracted away from the model. We pick back up post measurement, where a series of detections (no detections, the target detection, the target detection and clutterer detections, or just clutterer detections) are received and processed to produce a series of range and range-rates. These are processed in the Bayesian tracker, the BFP, where the optimal Bayesian estimate is computed using target dynamics, the previous state estimate and error covariance, the measurement model, and a weighted sum of the detections using PDA. This in turn feeds the error prediction block and the process continues until the track death, again a process not modeled here.

Chapter 3

WAVEFORM DESIGN

While the linear frequency-modulated (FM) chirp has excellent ambiguity properties, and allows for a large bandwidth to be compressed into a small time duration [15], it is not always the optimal signal with regard to minimizing the predicted mean-squared error (MSE). In [10], it was shown that in the dictionary of linear chirps constrained to a certain bandwidth and duration, it was sufficient to only keep the minimum and maximum duration chirp that utilizes the full bandwidth. However, by including time-frequency rotations of a signal using the fractional Fourier transform, the library out-performed the linear chirp only library [10]. This is because rotations of the time-frequency plane causes rotations in the ambiguity function domain, resulting in the projection of the ambiguity on the range or range-rate axes [24]. This can be used to allocate ambiguity or variance to dimensions where the tracker uncertainty is least great [4].

Nonlinear FM signals have been shown to be well-matched to wideband environments [4]. The *atan* chirp signal has been effectively used to suppress narrowband interference in ultrawideband systems [25]. In addition, generating FM signals in analog systems is possible through a series of cascaded integrators allowing for simple implementation with current technology [13]. We therefore aimed to create a relatively small dimension space over which to optimize, but with a rich set of nonlinear waveforms. Since nonlinear functions can be approximated by a truncated Maclaurin series [26], we start by introducing the cubic polynomial FM (CP-FM) signal instantaneous frequency:

$$f(t) = At^3 + Bt^2 + Ct \tag{3.1}$$

Equivalently, this results in the quartic polynomial phase function (QPPF):

$$\phi(t) = \frac{1}{4}At^4 + \frac{1}{3}Bt^3 + \frac{1}{2}Ct^2 + Dt \tag{3.2}$$

Note that any time-normalizing constants are omitted since the parameters (A, B, C) in general can contain the normalizing constants.

The instantaneous frequency of the CP-FM signal is characterized by the parameter set (A, B, C), for $A, B, C \in \mathbb{R}$. Depending on the values of these three parameters, a rich set of linear and nonlinear FM signals can be obtained. Specifically, the following signals can be represented:

- Rectangular pulse when $(A, B, C) = (0, 0, 0)$
- Linear chirp when $(A, B, C) = (0, 0, \pm 1)$
- Quadratic monomial chirp when $(A, B, C) = (0, \pm 1, 0)$
- Cubic monomial Chirp when $(A, B, C) = (\pm 1, 0, 0)$
- Arctangent chirp approximation when $(A, B, C) = (\frac{1}{3}, 0, 1)$
- Inverse arctangent chirp approximation when $(A, B, C) = (-\frac{1}{3}, 0, 1)$
- Exponential chirp approximation when $(A, B, C) = (\frac{1}{6}, \frac{1}{2}, 1)$

The approximated functions follow from the third order truncated Taylor series of the nonlinear functions about $t = 0$. This allows for waveform design in a continuous space dictionary (infinite possible waveforms) instead of a discrete space dictionary (finite number of possible waveforms). The independent time parameter of the cubic polynomial is constrained in magnitude to unity to coincide with most Taylor series' limits on accuracy. That is, most Taylor series for nonlinear functions of interest are only well-approximated for parameters constrained to unity in magnitude. If we allow for a higher dimensional parameter space, we allow for a more rich set of nonlinear frequency functions to be faithfully represented. However, the complexity may outweigh the benefit in that adding dimensions may hinder optimization. Cubic polynomials generally recreate shapes of interest, while avoiding stability issues with polynomials of high order.

There is a problem with this naive parameterization, however. Since the bandwidth and time duration of the signal scales and shifts the polynomial, there is some redundancy in the parameter space. For example, the waveform ultimately generated from the parameter coordinate (1, 1, 1) is the same as ($\frac{1}{3}$, $\frac{1}{3}$, $\frac{1}{3}$) after time and bandwidth normalization. To remedy this, we can constrain the coefficients such that the Euclidean norm is bound by unity:

$$(A^2 + B^2 + C^2)^{0.5} = 1 \tag{3.3}$$

This normalization amounts to constraining the coefficients to the surface of a three-dimensional unit sphere. This in turn reduces the dimensionality of search space to two dimensions since the surface of a sphere can be traversed using two angles for a fixed radius in spherical coordinates. This corresponds to the transformation [26]:

$$\begin{aligned} A &= \cos(\theta)\sin(\phi) \\ B &= \sin(\theta)\sin(\phi) \\ C &= \cos(\phi) \end{aligned} \tag{3.4}$$

where $0 \leq \theta \leq 2\pi$ and $0 \leq \phi \leq \pi$. Optimization methods now need only to explore a two-dimensional space, in addition to a time-frequency vector, since there is potential to optimize with respect to the time-bandwidth product [5]. In addition to the reduction in dimensionality, the search space has been stabilized by removing the origin. In the original unconstrained approach, small perturbations about the origin would be scaled and amplified when fitting to a particular pulse length and bandwidth. We therefore have two sets of parameters: shaping parameters (θ, ϕ), and time-bandwidth parameters (T, BW). We constrain 10 μs $\leq T \leq$ 100 μs and 1.5 MHz $\leq BW \leq$ 15 MHz.

We can further constrain this space to avoid including non-monotonic instantaneous frequency (IF) functions over the signal duration. This is because non-monotonic IF functions have repeated frequency components and oscillatory behavior. This could result in increased ambiguity in the Doppler domain, which is undesirable for radar waveform design. For cubic polynomials, we have at most one local maximum, and one local minimum. By forcing the CP-FM signal time support between these points, we can ensure the IF function remains monotonic. We can solve for these points by setting the first derivative equal to 0, or equivalently, looking for the roots of the quadratic polynomial resulting from taking the derivative of the IF of the CP-FM signal in Equation 3.1:

$$f'(t) = 3At^2 + 2Bt + C = 0 \tag{3.5}$$

$$t = \frac{-2B\pm(4B^2 - 12AC)^{0.5}}{6A} \tag{3.6}$$

For the case where the $4B^2 - 12AC > 0$, these roots are real and thus the local maxima and minima for the function are real and exist. When $4B^2 - 12AC = 0$, there is only one real root and thus only one inflection point for the cubic polynomial. Therefore, the polynomial is monotonic. When $4B^2 - 12AC < 0$, the roots are imaginary and thus the local maxima and minima for the function do not exist. Therefore, for parameters satisfying the inequality

$$4B^2 - 12AC \leq 0 \tag{3.7}$$

the function is monotonic. For regions in the parameter space that do not satisfy Equation 3.7, we must ensure that the signal duration in which we sample this polynomial is contained within the local maximum/minimum points. To do this, we can solve the inequality in the phase space

$$\cot^2(\phi) \leq \frac{3}{2}\sin(2\theta) \tag{3.8}$$

We can enforce these constraints in a multitude of ways. First, we can solve for these angles analytically and determine if they exist in contiguous regions of the angle space that lends themselves to simple exploration. We could also sample uniformly along the space and include only the points that satisfy Equation 3.8 in the grid-space. Finally, we can use Equation 3.8 as a rejection criteria for points in the optimization process. That is, the optimization process must iterate again or may be penalized by a cost function if it violates these constraints. The latter can be worked into the differential evolution framework, underscoring the simplicity of the method in the face of complex constraints.

Now that we have normalized the cubic polynomial space, we have two classes of parameters: shaping parameters, and time-bandwidth parameters. The shaping parameters consist of the angles ϕ and θ, and define the frequency function shape. By varying these parameters, linear FM chirps, quadratic FM chirps, or any FM signal with IF function consisting of a cubic polynomial over the unit time support can be created. The time-bandwidth parameters consist of the pulse duration and bandwidth. Once the shape is defined, the three coefficients of the polynomial IF can be found. Once they are found, the four coefficients of the phase function polynomial are easily found through symbolic integration and the well-known power rule for indefinite integrals [26].

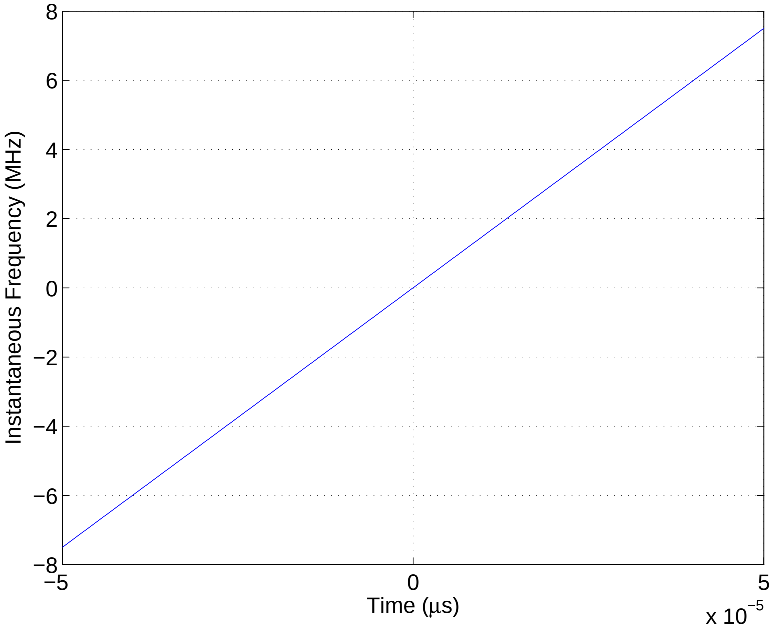

Figure 3.1: Linear

To map the polynomial, which is defined over the unit time support arbitrarily, to the pulse duration $[-T/2, T/2]$, we need to only scale the time axis. While this scaling is somewhat arbitrary, it provides key insight into the optimization process, and separability between the shaping parameters and the time-bandwidth parameters. This way, if an interesting shape is found, it is easy to explore what happens when only the time-bandwidth parameters are varied. To map over the baseband frequencies $[-BW/2, BW/2]$, the maximum and minimum values of the polynomial over $[-1, +1]$ must be found (assuming we have not yet scaled the time axis). This means we must solve for the critical points of an arbitrary third order polynomial. This is easily done by finding the points at which the derivative is zero. We can make some simplifying assumptions based on the coefficients.

In Figure 3.1, we have the case where only the first order coefficient is non-zero, $(A, B, C) = (0, 0, 1)$, and so the instantaneous frequency function is linear. In this case, there are no critical points, so we only need to look at the end points, $[-1, +1]$, for the minimum and maximum value. Depending on whether the slope is positive

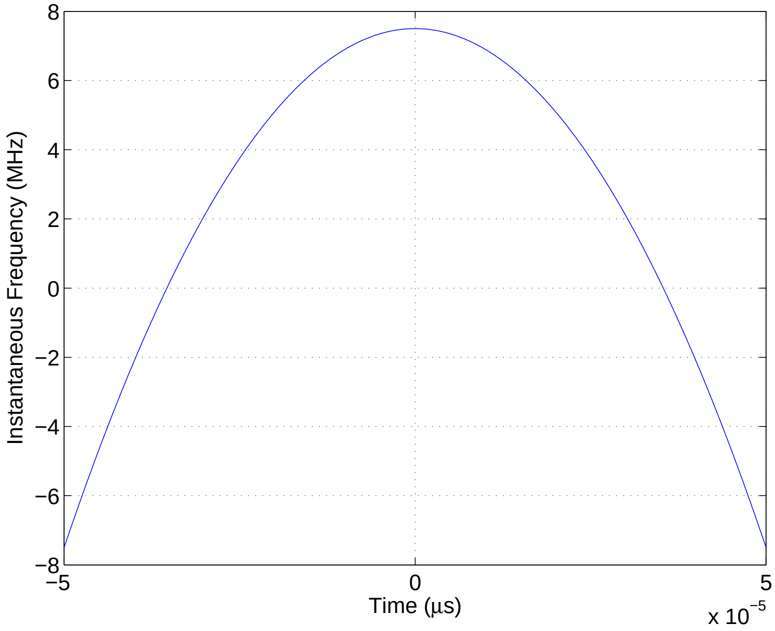

Figure 3.2: Quadratic

(upswept linear chirp), or negative (downswept linear chirp), the minimum and maximum correspond uniquely to one of the end points. Evaluating the cubic polynomial at these points, we then subtract the minimum value, divide all coefficients by the maximum value, multiply by the BW, and subtract $BW/2$. This centers the frequency function on 0 Hz, and allows it to span between $-BW/2$ and $BW/2$ for a passband bandwidth of BW. Then the time scale ensures it does this between $-T/2$ and $T/2$, and the transformation is complete. The trapezoidal envelope ensures the frequency excursion never exceeds the BW, and that the energy is not present outside of this time duration.

In Figure 3.2, we see the case where the second order coefficient is non-zero, $(A, B, C) = (0, -1, 0)$. We could in general have another critical point in addition to the end points. Here we see the critical point is at the time origin after normalization. Therefore the saddle points and the end points are all evaluated and a minimum and maximum search executed as before. Once this is found, the function is scaled and shifted as previously described.

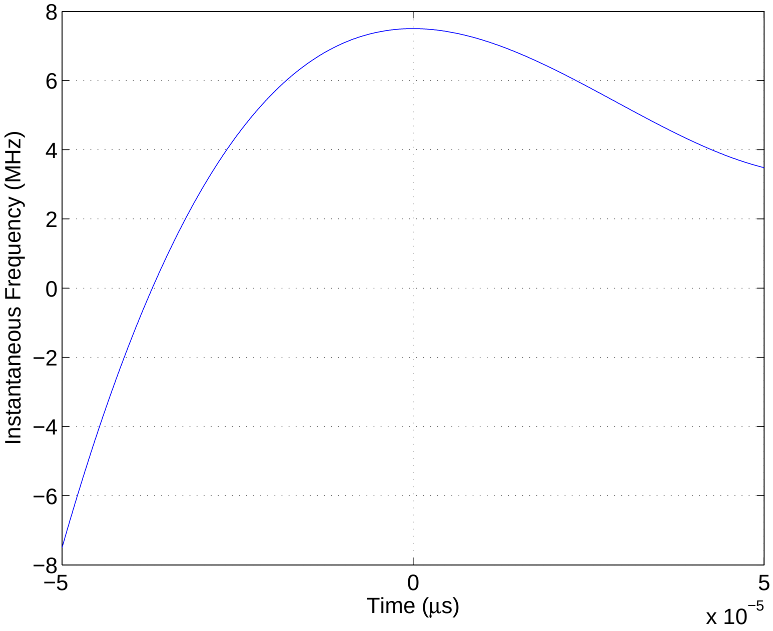

Figure 3.3: Cubic

Finally in Figure 3.3, an example of a third order cubic polynomial with $(A, B, C) = (\frac{1}{2}, \frac{-\sqrt{3}}{2}, 0)$ is shown. In these cases, we have the potential for 2 critical points in addition to the end points. In fact, we can clearly see that one of the critical points falls outside of the time window. This is why each critical point must be checked against the time bounds before including it in the scaling process. Again, the same scaling and shifting process is done with, at most, 4 critical points.

The process is summarized in Table 3.1.

```
Step 1: Select time-bandwidth parameters [T, BW]
Step 2: Select shaping parameters [$\theta$, $\phi$]
  A = cos($\theta$)sin($\phi$)
  B = sin($\theta$)sin($\phi$)
  C = cos($\phi$)
Step 3: Find critical points
  If (A $\neq$ 0) AND ($B^2 > 3AC$)
    Cp = [-1, +1, $\frac{-2B+\sqrt{4B^2-12AC}}{6A}$, $\frac{-2B-\sqrt{4B^2-12AC}}{6A}$]
  Else If (B $\neq$ 0)
    Cp = [-1, +1, -C/(2B)]
  Else
    Cp = [-1, +1]
Step 4: Remove critical points outside of [-1, +1]
Step 5: Find $y_{max}$ and $y_{min}$ by evaluating $y(x) = Ax^3 + Bx^2 + Cx$ for each x = Cp
Step 6: Scale and shift the polynomial
  $\tilde{A} = \frac{A(BW)}{(y_{max}-y_{min})(T/2)^3}$
  $\tilde{B} = \frac{B(BW)}{(y_{max}-y_{min})(T/2)^2}$
  $\tilde{C} = \frac{C(BW)}{(y_{max}-y_{min})(T/2)^1}$
  $\tilde{D} = \frac{-y_{min}(BW)}{(y_{max}-y_{min})} - BW/2$
  $\tilde{y}(x) = \tilde{A}x^3 + \tilde{B}x^2 + \tilde{C}x + \tilde{D}$
```

Table 3.1: QPPF normalization algorithm

Chapter 4

WAVEFORM SELECTION

Gradient search methods for minimizing the predicted mean-squared error (MSE) (PMSE) were shown to be prohibitively expensive [5]. In addition, to date the only tractable solutions have been limited to linear chirp functions for their search space and suffered from sensitivity to differentiation noise [5]. The use of genetic algorithms for waveform design has been investigated in [12]. However, the ambiguity function was used for the fitness function, and therefore optimization was not directly attempting to minimize the PMSE for tracking. In addition, the family of waveforms was restricted to bi-phase and poly-phase codes that do not contain non-linear frequency-modulated (FM) signals that are often of interest in cluttered and wideband environments [4]. The goal of [12] was to find binary coded sequences when Barker codes, which have the lowest peak-to-sidelobe ratios for non-periodic codes [2], did not exist [12]. For example, there are no known Barker codes of length greater than 13 [2], and a length 13 Barker code for a bistatic radar is no longer optimal since the ambiguity function dynamics change [12]. To this end, it was shown that genetic algorithms can generate binary coded sequences with good ambiguity sidelobe properties with little effort for scenarios where the optimal codes do not exist.

The benefit to using genetic algorithms is illustrated even in this simplified case. For example, the space of quad-phase length 16 codes, which was explored in this case, contains $4^{16} = 4,294,967,296$ possible waveforms. In 100 generations, they were able to find a length 16 code with superior integrated sidelobe level to a popular length 16 code called the Frank 16 code. However, while this demonstrated the power of genetic algorithms for waveform design, it is extremely limited given that only binary

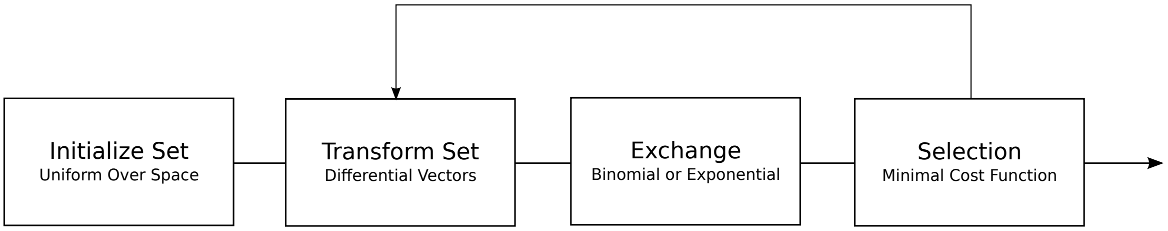

Figure 4.1: DE algorithm sequence

coded sequences are considered. Further, the parameters of the genetic algorithm were not specified. The toolbox used was detailed in a technical report [27], and the algorithm defined is similar to differential evolution (DE) in some aspects, but different in many others. In addition, the waveform design was ultimately fixed, and did not vary with time. As previously mentioned, other efforts to develop optimal waveforms has been limited to simplified models or assumptions, such as a library of linear chirps or time-frequency rotated waveforms [10]. However, optimal waveforms not included in dictionaries are never selected, and increasing the size of libraries linearly increases the search time. Previous approaches to track-before-detect using waveform optimization have also been limited to linear chirp libraries [7].

DE is a genetic algorithm that allows for efficient optimization in complex environments [28]. It works by perturbing, combining, and comparing the cost of a parameter vector. Due to its simple but powerful nature, no differentiation is required and it subsequently does not have any of the limitations of gradient search methods. Unlike dictionary based methods, it still can traverse a continuous parameter space. In Figure 4.1, the top-level algorithm for DE is shown [28]. The population (set) consists of individual vectors in the parameter space to optimize, and these vectors are called genomes [28]. To simplify the rhetoric for this thesis, we make adaptations to the terms used in the DE process. We therefore refer to genomes as candidates. The candidates are randomly and uniformly distributed over the waveform space. After the initialization, each candidate's cost is evaluated. The goal is to minimize this cost function. The number of candidates is a parameter in the optimization process, N_p.

After initialization, a mutation step occurs. The mutation is difference-based, hence the name *differential* evolution. For each set member, a scaled-difference transform (SDT) is performed. The candidate being transformed is called the target vector. The scaled difference of two other vectors in the set, unique from each other and unique from the target vector, is added to the base vector. The base vector is random for 'rand' mode, the best (minimum cost) in 'best' mode, and weighted with the target and best vector in 'target-to-best' mode. The SDT step is configurable in two ways. The scaling factor on the differential, F, typically chosen between [0.4, 1] affects how much the difference pair vector contributes to the transformed vector. The number of difference pairs is also configurable; as pairs are added, each differential pair must be unique amongst the total set of vectors being added (that is, they cannot be the target vector, the base vector, or any vectors used in any other candidate vectors, and unique amongst one another). The summation of the target vector and the difference vectors is called the donor vector in DE literature.

After we have obtained the transformed vectors spawned from each of the candidates in the set, the candidates undergo crossover, sometimes called recombination. We refer to it here as exchanging. This is where the candidates probabilistically exchange components from their respective SDT vectors. Two types of exchange are used: binomial, and exponential. For each, there is a statistical likelihood that a dimension of the SDT vector replaces a component of the candidate vector. For example, if the optimization space had dimension 2, then the exchange probability would be the probability that the first dimension of the candidate vector is replaced with the first dimension of the SDT vector. This is repeated for the second dimension as an independent trial. This probability distribution follows the namesake, either exponential or binomial. The vectors resulting from the exchange are called the trial vectors, which we subsequently refer to as new candidates.

Finally, the new candidates' cost are computed and compared with the original candidate set's respective costs. If the cost of the new candidate is less than or equal to the cost of the original candidate, then the new vector replaces the old vector in the set. The reason we choose the new vector in the case of equality in the cost is to promote exploration of the search space and avoid local minima. The process then starts over with the new set (containing some of the old potentially, and some of the new potentially) by transforming again and repeating. This is repeated for G_{max} times, the total number of generations (iterations), or until some cost criteria is met. The best (lowest cost function) vector in the set after the last selection step is the chosen vector for the minimization of the problem. There is no guarantee of convergence to the global minimum, and so this may not be the vector that minimizes the cost function. The convergence depends on the cost function, the exchange rate (Cr), the scaling factor (F), the number of set members (N_P), and number of iterations (G_{max}).

An example of the optimization process is shown in Figure 4.2. Here we stop and plot a point for each selected candidate. In this example, we have chosen DE/best/1/bin as the optimizer, and set size $N_P = 20$, $F = 0.5$, and exchange rate $Cr = 0.5$. The true minimum is shown by the large red X. Here we are minimizing Beale's function [29], a common optimization function, whose cost landscape is shown in Figure 4.3. In the complete limits of the function, the complicated optimization surface is hard to see. Another way to see the complete landscape is to compress the visual dynamic range with a log scale as shown in Figure 4.4. Here, the minimum is much more visible.

In this case, we set a stopping criterion to within a certain MSE since the true minimum is known. In practice, we terminate after a predetermined number of iterations since the true minimum PMSE is not known. This is a parameter to tune, as

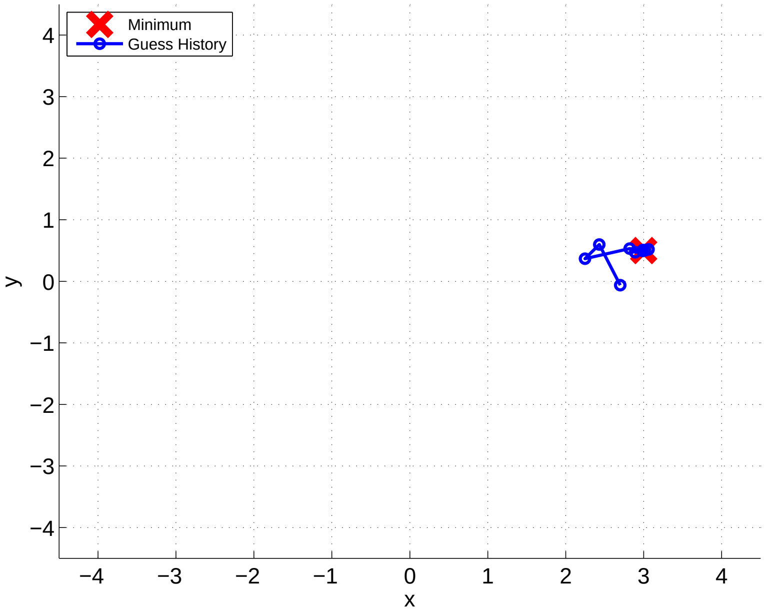

Figure 4.2: DE optimization example of Beale's function

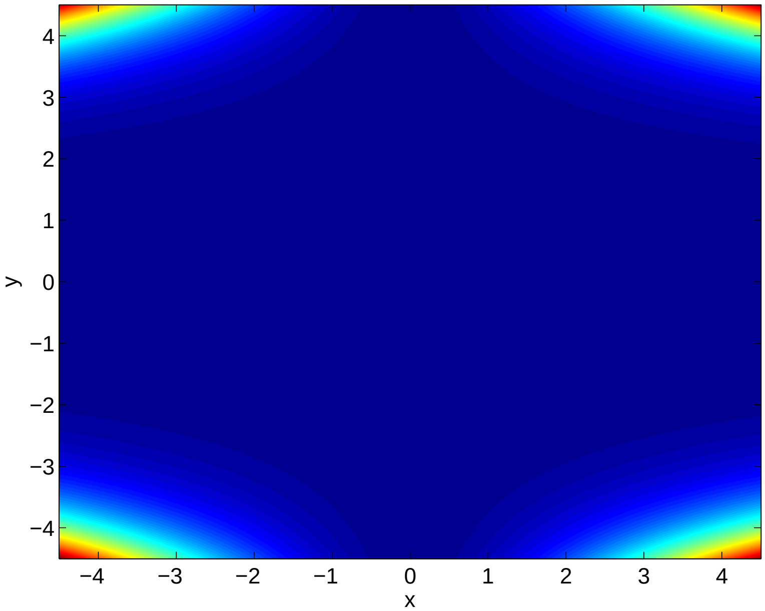

Figure 4.3: Beale's function

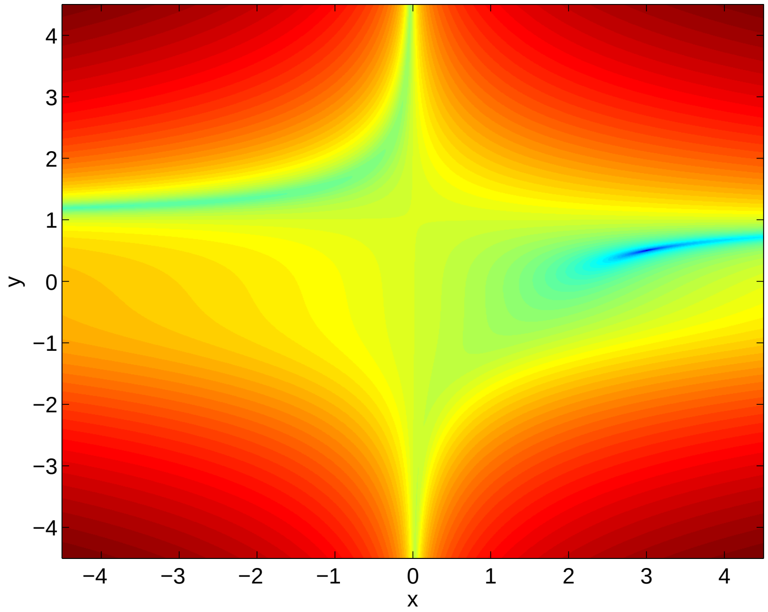

Figure 4.4: Beale's function, logrithmic scale

there is a clear tradeoff. The more iterations, the better the chance of approaching the true minimum. However, the additional cost and time to process after a certain point may yield diminishing returns.

To explore the tradeoffs of the various parameters, we tested the DE optimization engine on Beale's function using a variety of settings. Table 4.1 shows the results of this test. It is clear that the number of iterations required depends heavily on the various parameters. In [28], various researchers have provided starting points for these settings depending on the parameters of the optimization problem. The notation used, for example DE/best/1/bin, is shorthand for conveying the optimization settings [28]. The acronym DE is used to indicate differential evolution is being used. The second argument indicates the type of base vector used in the SDT. The third argument indicates the number of differential pairs to use in the SDT, typically 1 or 2. The fourth argument indicates the exchange type, either bin for binomial or exp for exponential.

Type	NP	F	Cr	Mean Iterations	Divergence Rate
DE/best/1/bin	20	0.5	0.5	26	.19
DE/best/1/bin	50	0.5	0.5	23	.10
DE/best/1/bin	20	0.9	0.5	51	.06
DE/best/1/bin	20	0.5	0.9	18	.22
DE/rand/1/bin	20	0.5	0.5	54	.00
DE/target-to-best/1/bin	20	0.5	0.5	44	.12
DE/best/2/bin	20	0.5	0.5	41	.09
DE/best/1/exp	20	0.5	0.5	26	.21

Table 4.1: Optimization results of Beale's function to within -70 dB MSE of the true minimum

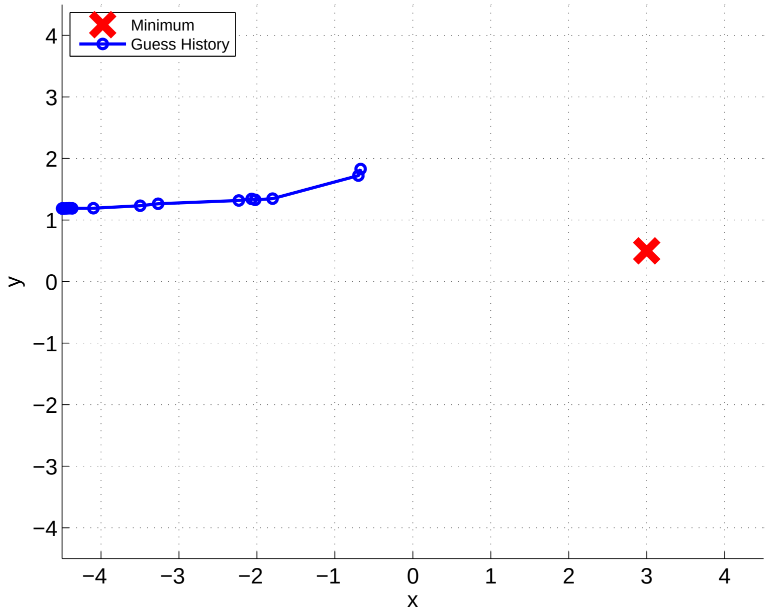

Figure 4.5: DE optimization example of Beale's function, divergent track

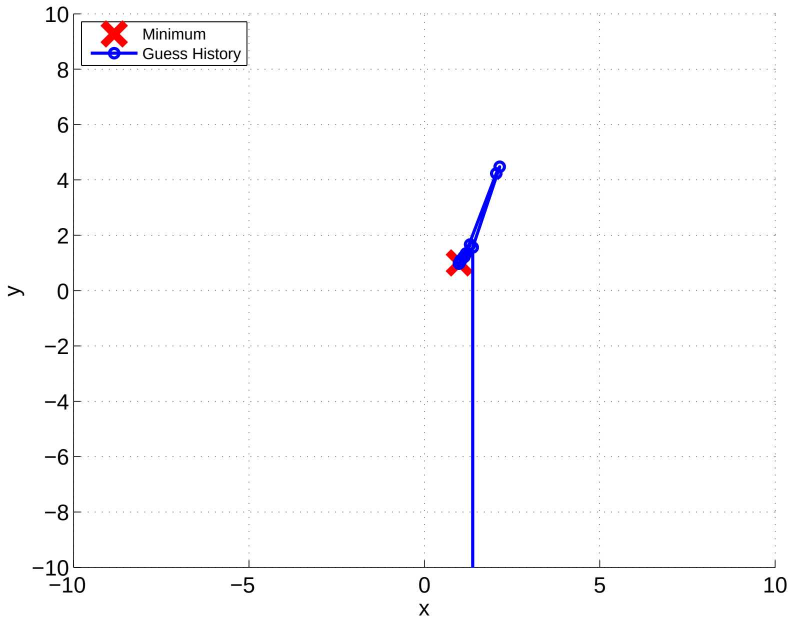

Figure 4.6: DE optimization example of the Rosenbrock function, $n = 1$

One subtlety not yet discussed is that these evolved targets could potentially fall outside of the limitations of the search space. For the example with Beale's function, the function grows unbounded away from the minimum, and so it is often constrained to x = [-4.5, 4.5], as we have done here. With differential evolution, the initialization is done by randomly spreading candidates uniformly over the search space. If this is unbounded, then we must guess as to what the correct range is. For example, the Rosenbrock function is used for unconstrained optimization problems [29], and while the minimum is at [1, 1], if this is not known *a priori*, an estimate of where to contain the minimal point must be made. If the minimum is not contained within this initial space, it can still be discovered in subsequent iterations, but convergence may be slow [28].

Figure 4.6 shows the optimization of the Rosenbrock function for $n = 1$ (so 2 dimensions) [29]. In this example, the initial set was spread over the 200x200 square bounded by [-100 -100] and [100 100] for the minimum and maximum parameters, respectively. As a result, the algorithm tracks away from the minimum for a few

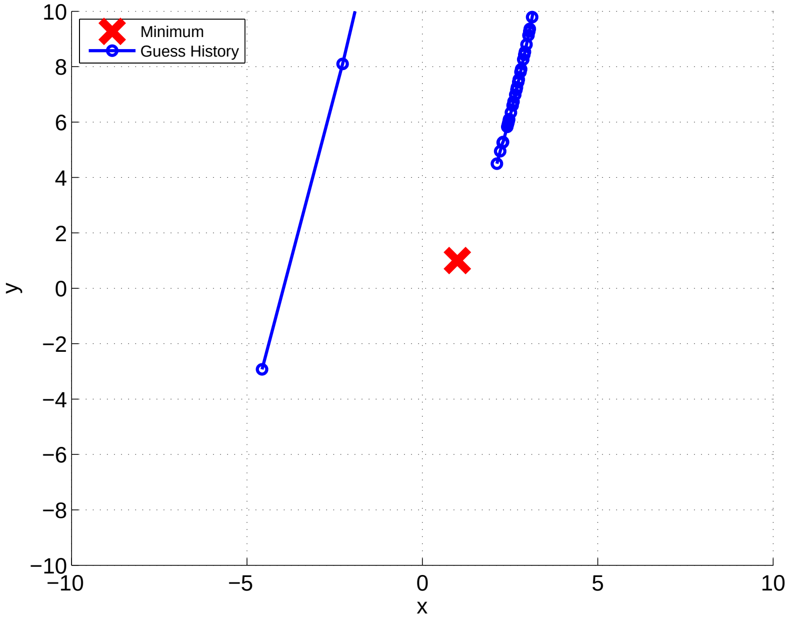

Figure 4.7: DE optimization example of the Rosenbrock function, $n = 1$, divergent track

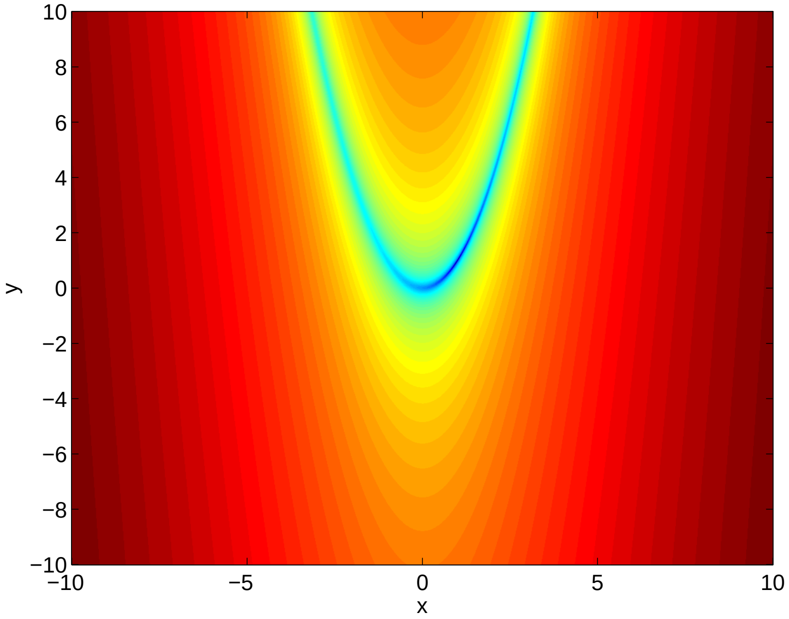

Figure 4.8: Rosenbrock function, $n = 1$, log scale

Type	NP	F	Cr	Mean Generations	Divergence Rate
DE/best/1/bin	20	0.5	0.5	53	.52
DE/best/1/bin	50	0.5	0.5	50	.16
DE/best/1/bin	20	0.9	0.5	88	.75
DE/best/1/bin	20	0.5	0.9	28	.80
DE/rand/1/bin	20	0.5	0.5	97	.98
DE/target-to-best/1/bin	20	0.5	0.5	98	.99
DE/best/2/bin	20	0.5	0.5	86	.64
DE/best/1/exp	20	0.5	0.5	61	.56

Table 4.2: Optimization results of the Rosenbrock function $n = 1$ to within -70 dB MSE of the true minimum, initial search over [-100 -100] [100 100]

trials, then slowly makes its way back. The actual error landscape is shown on a log scale in Figure 4.8.

The results in Table 4.2 and Table 4.3 are interesting in a multitude of ways. First we note that the configurations best suited for Beale's function were not necessarily best suited for this function. There have been many attempts to capture and recognize which configurations work for what type of functions and parameters, but the overall consensus is no general solution works the best all the time [28]. There has been work done in randomly generating these parameters with each time step, specifically for dynamic landscapes [30]. Part of the results is to find which configurations appear to converge to a lower MSE in a shorter period of time.

The other interesting, albeit expected result is that constraining the search space has benefits in terms of convergence rate. Constraining the instantaneous frequency (IF) polynomial coefficients to the surface of the unit sphere is possible with DE and

Type	**NP**	**F**	**Cr**	**Mean** **Generations**	**Divergence** **Rate**
DE/best/1/bin	20	0.5	0.5	44	.04
DE/best/1/bin	50	0.5	0.5	33	.00
DE/best/1/bin	20	0.9	0.5	80	.09
DE/best/1/bin	20	0.5	0.9	23	.46
DE/rand/1/bin	20	0.5	0.5	82	.30
DE/target-to-best/1/bin	20	0.5	0.5	82	.46
DE/best/2/bin	20	0.5	0.5	70	.11
DE/best/1/exp	20	0.5	0.5	46	.06

Table 4.3: Optimization results of the Rosenbrock function $n = 1$ to within -70 dB MSE of the true minimum, initial search over [-10 -10] [10 10]

has a variety of methods that can be used such as penalties, reinitialization, and bounce back [28]. Allowing the spherical angles to overflow their angle may have optimization benefits as well in encouraging global minimum searches. Though DE can be slow to converge [12], we have the ability to further constrain the search space to speed up convergence. For example, oscillatory FM signals may be undesired and so the DE algorithm can be bounded in the spherical domain to restrict angles that would produce this frequency function.

At each time step, the optimization landscape changes as the target, which we assume is in motion, traverses along its trajectory. As a result, we have a few options as to how to resume the optimization method. One method is to simply start over [28]. Another method is to perturb the candidates about their current location and reevaluate the cost function [30]. However, this method typically assumes knowledge of the modality, that is the number of peaks currently present in the cost landscape.

The dimensionality of the optimization space is a minimum of 2, encompassing the two angles required to traverse the surface of the unit sphere as discussed in Chapter 3. In this case, the bandwidth and signal duration are fixed to some value. In other scenarios, we let these parameters be a part of the optimal search space since maximizing the time-bandwidth product is not always optimal [4]. In this case, the DE engine is configured for a four-dimensional (4D) search space. Unlike the angles which wrap naturally at their boundaries, the bandwidth and signal duration must be constrained. In order to prevent the optimization engine from selecting out of bound parameters, we can impose a brickwall penalty [28]. This means that the cost function outputs an extremely large value for parameters that are out of bounds to guarantee they are not picked for selection. This is not an issue with initialization since we can bound the uniform random distribution of initial points. However, as the candidates transform and exchange, they may explore outside of the physically limited parameter space. The SDT control factors can also be tuned to attempt to avoid overzealous exploration that would motivate more candidates to violate the constraints of the system. However, this sort of tuning can cause slower convergence by limiting exploration within the valid regions. This tradeoff is explored. In addition, we could also saturate the values, so attempted new candidates that exceed the minimum BW could be limited to the minimum BW. New candidates that exceed the maximum BW would be limited to the maximum BW, and similarly for the time duration. While this helps with exploration of the limits, it may favor or bias the edges of the time-bandwidth space.

Since it is not known precisely what the minimum predicted MSE truly is at any time step, we simply tune with various simulations the number of iterations, G_{max}. This has a direct impact on the minimal value achieved, as well as the aggregate computational complexity since more generations require more evaluations of the cost function. The complete DE algorithm for this thesis is outlined in Table 4.4. Note that while the distribution for the exponential is called a geometric, it is actually a truncated geometric distribution with parameter $(1-Cr)$. It can be generated using a geometric distribution and setting all answers larger than 4 equal to 4. Alternatively, it has been generated [28] by setting $L = 1$, drawing a uniform random variable, and incrementing L while rand $\leq Cr$ and $L < 4$.

```
Step 1: Uniformly and randomly select $N_p$ candidates
    $g_k(1) \sim U(0, 2\pi)$
    $g_k(2) \sim U(0, \pi)$
    $g_k(3) \sim U(T_{min}, T_{max})$
    $g_k(4) \sim U(BW_{min}, BW_{max})$
Step 2: Calculate the cost of the candidate set
Step 3: Form SDT set by transforming candidates
    If (‘rand’)
        $\mathbf{d}_k = \mathbf{g}_l + F(\mathbf{g}_i - \mathbf{g}_j)$, $i \neq j \neq k \neq l$
    If (‘best’)
        $\mathbf{d}_k = \mathbf{g}_{best} + F(\mathbf{g}_i - \mathbf{g}_j)$, $i \neq j \neq k$
    If (‘target-to-best’)
        $\mathbf{d}_k = \mathbf{g}_k + F(\mathbf{g}_{best} - \mathbf{g}_k) + F(\mathbf{g}_i - \mathbf{g}_j)$, $i \neq j \neq k$
Step 4: Exchange transformed set with candidate set to produce new candidates
    If (‘bin’)
        $j_{rand} \sim U[\{1, 2, 3, 4\}]$
        $g_k(i) = d_k(i)$ If i = $j_{rand}$ OR $rand_{k,i} \leq Cr$
    If (‘exp’)
        $n \sim U[\{1, 2, 3, 4\}]$, $L \sim Geometric[\{1, 2, 3, 4\}]$
        $g_k(i) = d_k(i)$ For i = n, n+1, ..., n+L (mod 4)
Step 5: Compute cost of new candidate set
Step 6: Select old candidates or new candidates based on cost
    If ($Cost(\mathbf{d}_k) <= Cost(\mathbf{g}_k)$)
        $\mathbf{g}_k = \mathbf{d}_k$
Step 7: Stop if $G_{max}$ iterations achieved or go back to Step 3
$\mathbf{DE/[rand|best|target\text{-}to\text{-}best]/[1|2]/[bin|exp], N}_p\mathbf{, Cr, G}_{max}$
```

Table 4.4: DE algorithm

Chapter 5

RESULTS

Each of the results were obtained by running the simulation for 1,000 convergent trials. In order to categorize tracks as divergent or convergent in an automated fashion, if the actual normalized mean-squared error (MSE) exceeded -30 dB, the track was discarded. Therefore, if 50% of the tracks diverged, then a total of 2,000 trials would be needed to achieve 1,000 convergent tracks. From this ratio, the convergence rate is calculated, and from the 1,000 convergent tracks, all of the average statistics were computed.

It was found that particle degeneracy was a problem. Some exploration into the results found it was due to the likelihood being ill-conditioned, and so particle collapse was common. Increasing the number of particles helped, but became prohibitively expensive without solving the problem in general. Hand-tuning of the likelihood scaling, specifically the measurement covariance, remedied the problem, but is obviously undesirable since the hand-tuned solution required can vary from situation to situation. The problem ultimately was the high signal-to-noise ratio (SNR) assumption. Targets were initially chosen to have SNR on the order of 20+ dB. It is a well-known problem that high SNR tracking can cause a peaked likelihood, one that is extremely difficult to sample from with the transition prior [31]. The unscented particle filter (UPF) was proposed in [31] to remedy the problem, but with great additional computational complexity, since each particle must be propagated through the unscented transform (UT). Instead, moving the sensor relative to the target to put the SNR on the order of 10-18 dB sufficiently self-conditioned the likelihood, and prevented excessive particle collapse. This is still relatively high SNR, high enough

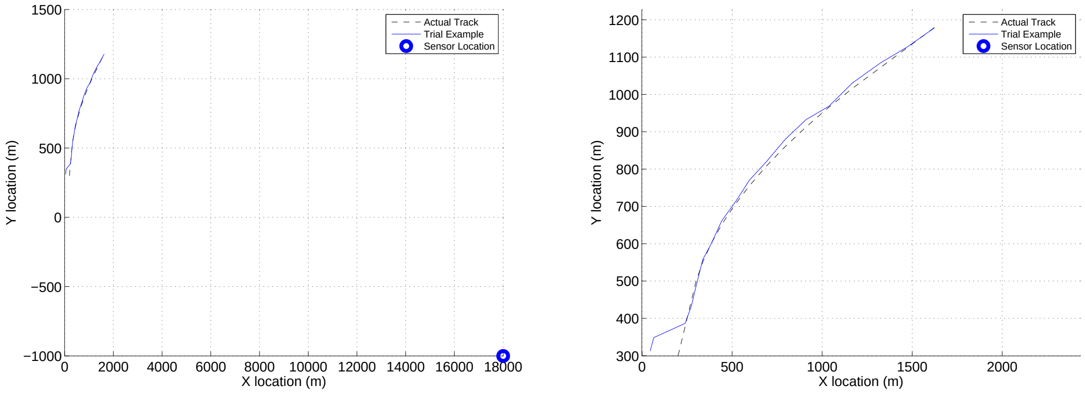

Figure 5.1: Actual track with example simulation overlay

to ignore the higher order noise terms in the actual likelihood function [15].

The first scenario we look at is shown in Figure 5. This track is an accelerating turn with an initial uncertainty, with the sensor placed near the radial center of the turn ellipse. As previously mentioned, the distance was chosen to provide high enough SNR for $P_D > 0.95$, but not so high to create a peaked likelihood and poor particle filter performance [31]. The noise intensity q_k was set to 2500 to accommodate the large initial uncertainty and to account for the non-constant velocity. The initial point was picked by perturbing the mean with noise of covariance $\text{diag}([10000, 500, 10000, 500])$, where $\text{diag}(x)$ denotes a diagonal matrix with diagonal elements equal to the elements of vector x. The cost function weighting was equal between range and range rate, or $\text{diag}([1, \Delta t^2, 1, \Delta t^2])$. The sensor was located at $(x, y) =$ (18000, -1000), and the bearing measurement had a variance of 10^{-6}. The bootstrap particle filter (BFP) with 1,000 particles was used as the tracker. Finally, the clutter density ρ was set to 10^{-5}.

We start by comparing commonly chosen DE parameters [28], DE/best/1/bin, to the four extremes of the time-bandwidth space using a linear chirp space in Table 5.1. This notation is used as shorthand to indicate DE is used with the 'best' base vector, one difference pair, and binary exchange. As we can see, the DE optimized

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	39
▶ Upswept Linear Chirp (BW_{min},T_{min})	54
▶ Upswept Linear Chirp (BW_{max},T_{min})	9
▶ Upswept Linear Chirp (BW_{min},T_{max})	81
▶ Upswept Linear Chirp (BW_{max},T_{max})	42

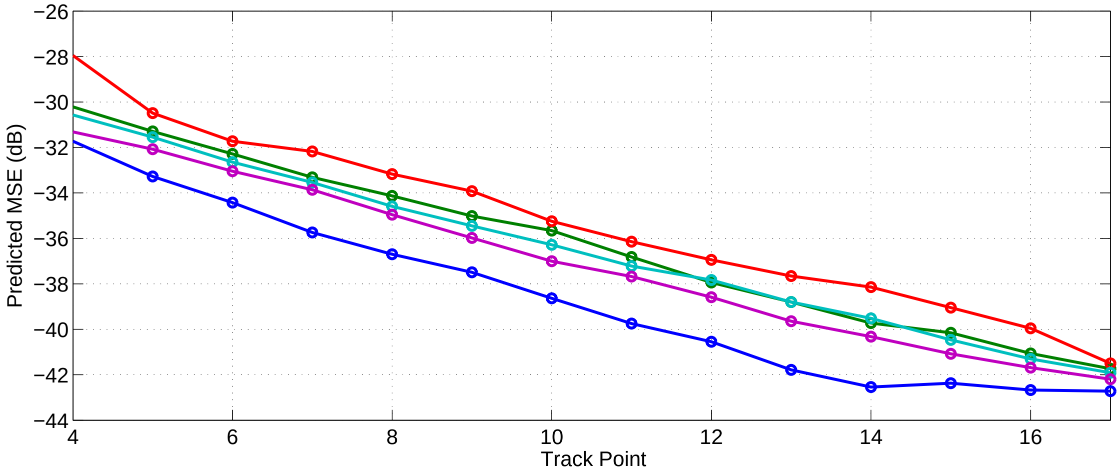

Table 5.1: Low clutter, differential evolution (DE) vs. upswept linear chirps

configuration outperformed all of the upswept linear chirps, some by as much as 4.5 dB at certain points in the track. Note that we have pruned the first 3 points of the track which represent the period in which the track was being acquired, as they were subject to large variations in error with little value to the predicted MSE (PMSE) of the converged track. An important element to note is that we have shown the convergence rate for each waveform as well. While we are minimizing the PMSE, it is interesting to note the other waveforms tended to have a higher convergence rate, with exception of the maximum bandwidth, minimum time duration upswept linear chirp.

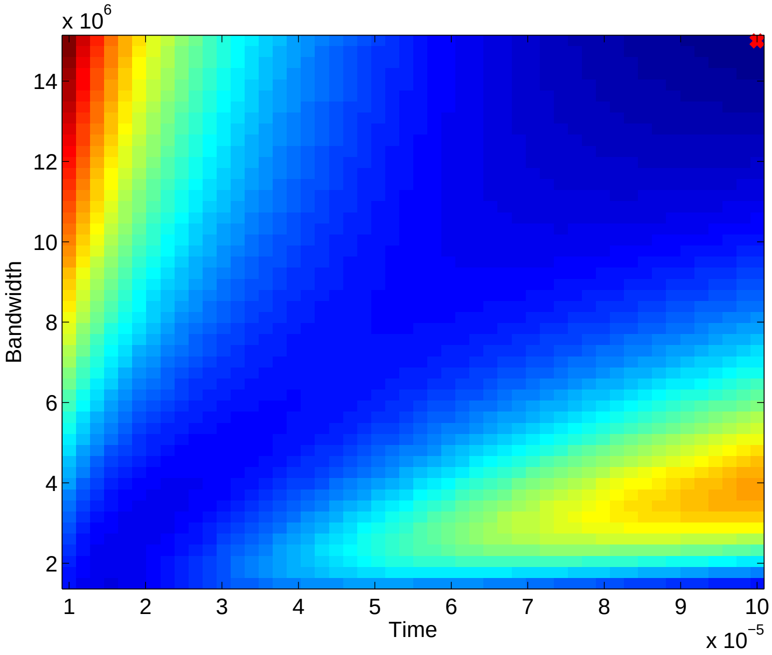

Figure 5.2: Error surface two-dimensional (2D) cut for fixed shaping parameters

To motivate the optimization method, an example of the time-bandwidth error surface for a fixed shaping parameter cut surface is shown in Figure 5.2. Clearly, since the surface in this example in non-convex, the function is non-convex in general. This is why gradient methods are ill-suited, even if they were computationally feasible, which is what motivated grid-based methods in [3]. Note the error surface in Figure 5.2 was generated using a highly sampled grid over the 2D time-bandwidth (TB) cut surface. The minimum is marked by a red X. Grid-based methods could also be susceptible to choosing local minima as well. If the error surface well containing the true minimum is not sampled in the grid, even if refined grid-methods are used (such as resampling the grid based on the first coarse grid iteration), the refined grid may not sample the true minimum or surrounding points along the minimum contour. This is where DE is well suited to optimize, as non-convex and complicated error surfaces are one of the primary motivators behind its use [28]. Specifically, the adaptive nature of DE leads to a phenomenon called contour matching, where error surface wells are automatically explored when detected through the iterations. In

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	39
▶ Downswept Linear Chirp (BW_{min},T_{min})	49
▶ Downswept Linear Chirp (BW_{max},T_{min})	8
▶ Downswept Linear Chirp (BW_{min},T_{max})	46
▶ Downswept Linear Chirp (BW_{max},T_{max})	11

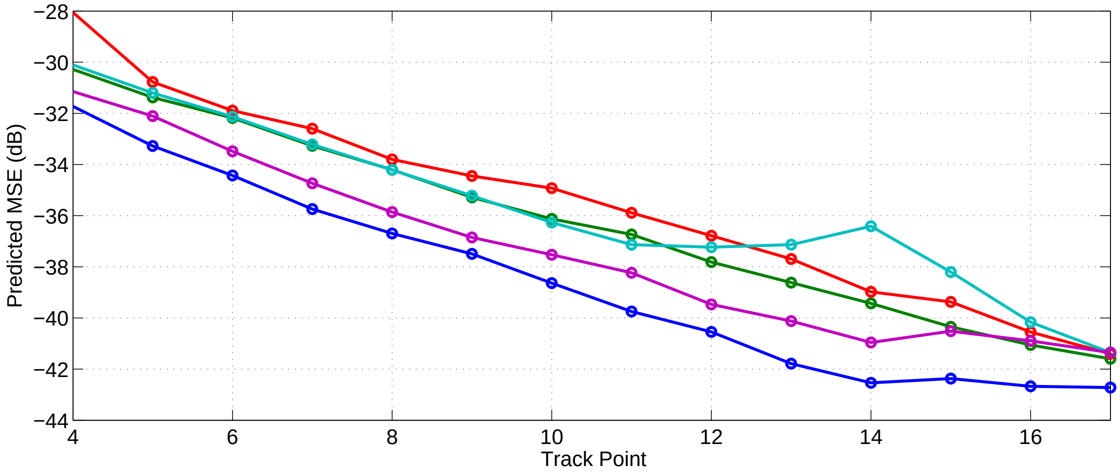

Table 5.2: Low clutter, DE vs. downswept linear chirps

addition, mutations allow for exploration outside of current candidate populations which means with each trial it is less likely to be stuck in a local minimum. However, this stochastic dependence is why the number of iterations is critical, as there must be enough to allow for a high probability that the global maximum was found.

In Table 5.2, we look at the downswept linear chirp library in contrast to the DE baseline. Again, the configured waveform outperforms the fixed library, this time as much as 6 dB. Interestingly, the convergence rate gap closed over the upswept chirps for this particular track. It also seems for this track, the maximum TB waveforms are heavily favored. While it was hypothesized in Chapter 3 that monotonicity would be a desirable property for ambiguity, we found that the most commonly chosen

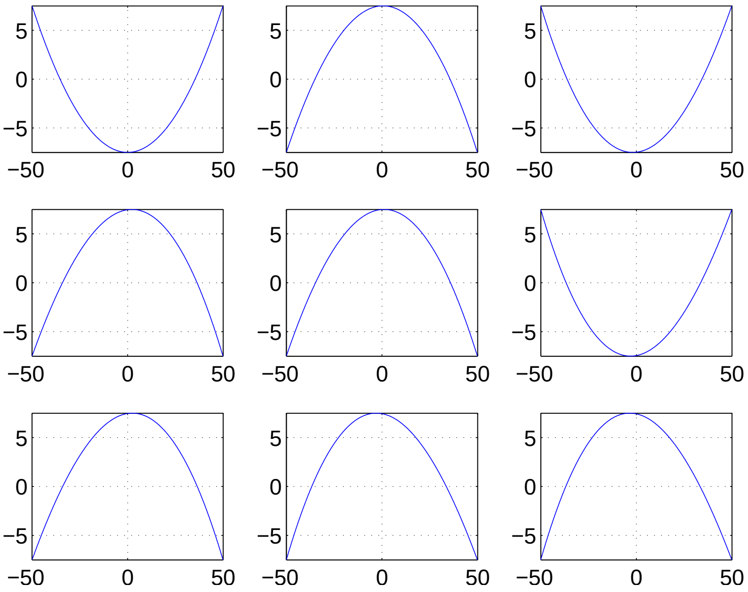

Figure 5.3: Typical instantaneous frequency (IF) for each track point (1-9); horizontal labels are time in μs and the vertical labels are frequency in MHz

shape for this track was the quadratic parabolic chirp (QPC) as shown in Figure 5.3. Therefore, constraining for monotonicity appeared to have little benefit and was not subsequently explored. To explore this waveform more in depth, we tested the 4 corners of the time-bandwidth cube much like the linear chirp, but with the QPC shaping parameters $(\theta, \phi) = (\frac{3\pi}{2}, -\frac{3\pi}{2})$ in Table 5.3. The IF function of the QPC is shown in Figure 5.4. Here, we finally see a fixed waveform that performs nearly as well as the optimized, a not surprising result since this particular waveform appears to be heavily favored for this scenario. One thing worth noting is that while the PMSE is nearly identical for the DE optimized configuration and the fixed QPC with the maximum TB product, the convergence rate is higher for the configured waveform.

To investigate the QPC further, we first look at the spectrogram in Figure 5.5. This provides some interesting insight, as the color intensity is indicative of the degree of spectral content around those frequency bands due to saddle point of the quadratic IF function. Therefore, we expect to see the frequency domain peaked at this location

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	39
▶ Quadratic Parabolic Chirp (BW_{min},T_{min})	56
▶ Quadratic Parabolic Chirp (BW_{max},T_{min})	38
▶ Quadratic Parabolic Chirp (BW_{min},T_{max})	53
▶ Quadratic Parabolic Chirp (BW_{max},T_{max})	36

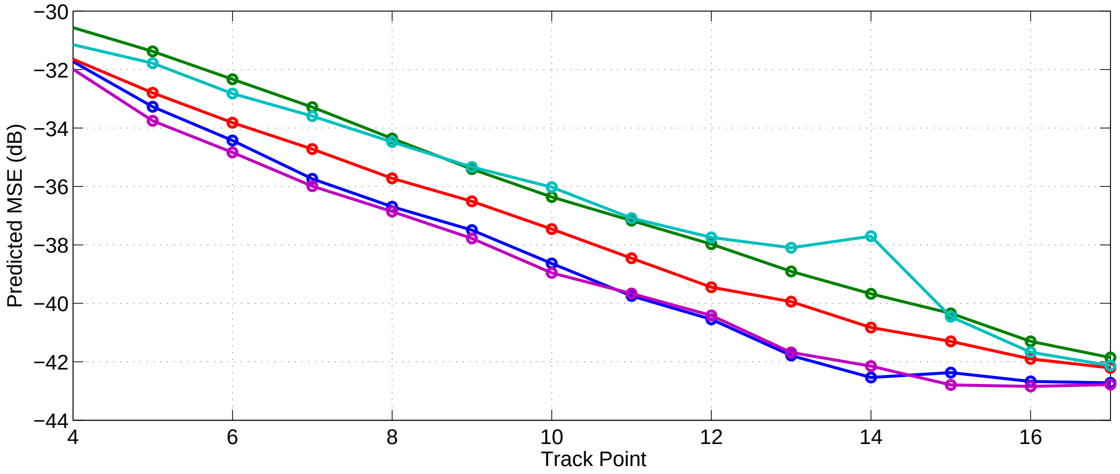

Table 5.3: Low clutter, DE vs. QPCs

instead of say a linear chirp where the frequencies are equally weighted throughout the IF band. As we can see in Figure 5.6, this is precisely the case. Interestingly, since the function is centered around 0 Hz, this peak is offset from 0 Hz by $BW/2$, in this case 7.5 MHz. The behavior here is highly oscillatory, much like a linear chirp's spectrum. However, since the receiver would ultimately have a matched filter, the Fourier transform of the square of the pulse represents the zero-lag cut of the ambiguity function. We see this in Figure 5.7. This is a big difference, as the matched filter has caused a dramatic peak around 0 Hz, with very little smearing. This is highly desirable, as there is little Doppler ambiguity to potentially perturb the peak. The next step is to look at the zero-Doppler cut by computing the autocorrelation

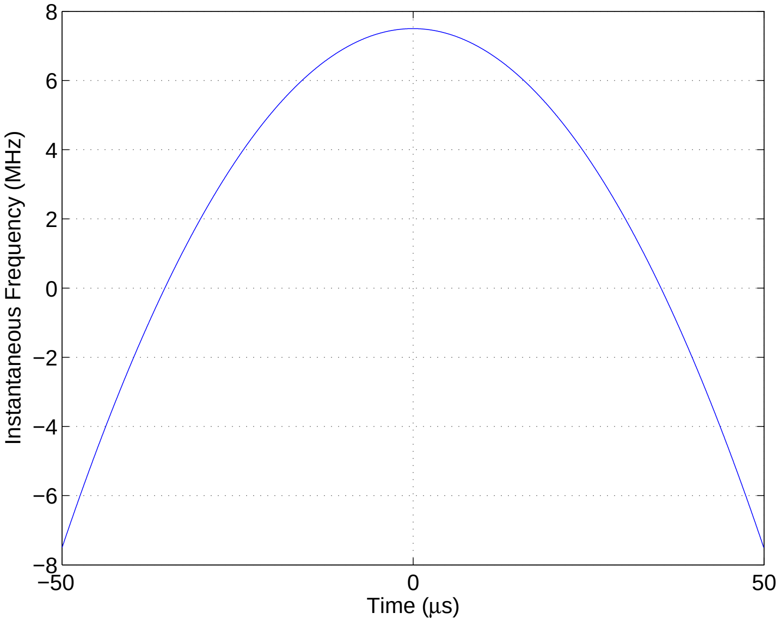

Figure 5.4: IF for the QPC

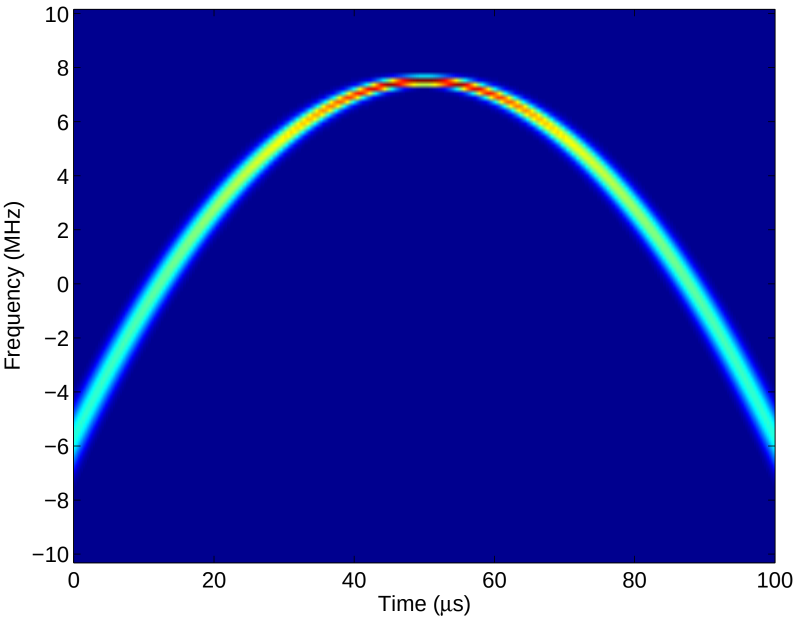

Figure 5.5: Spectrogram for the QPC

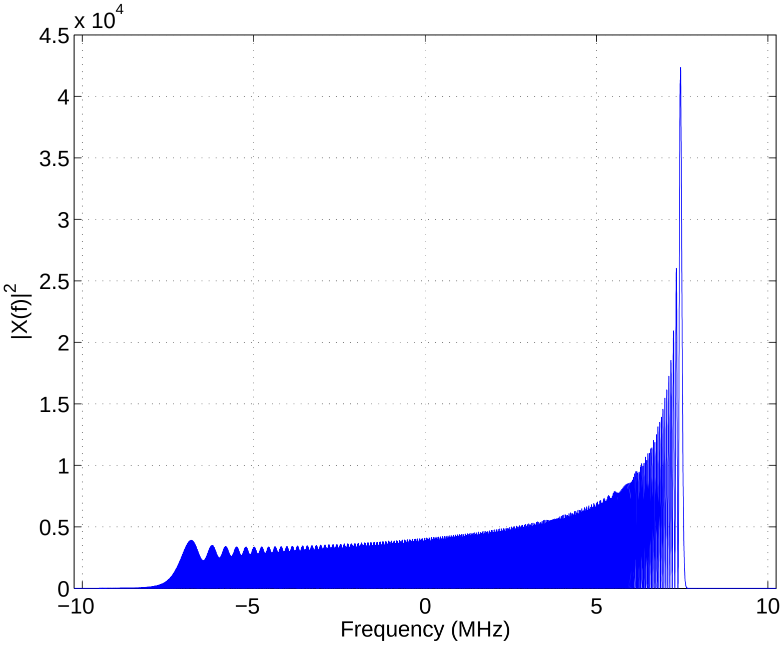

Figure 5.6: Fourier transform of the QPC

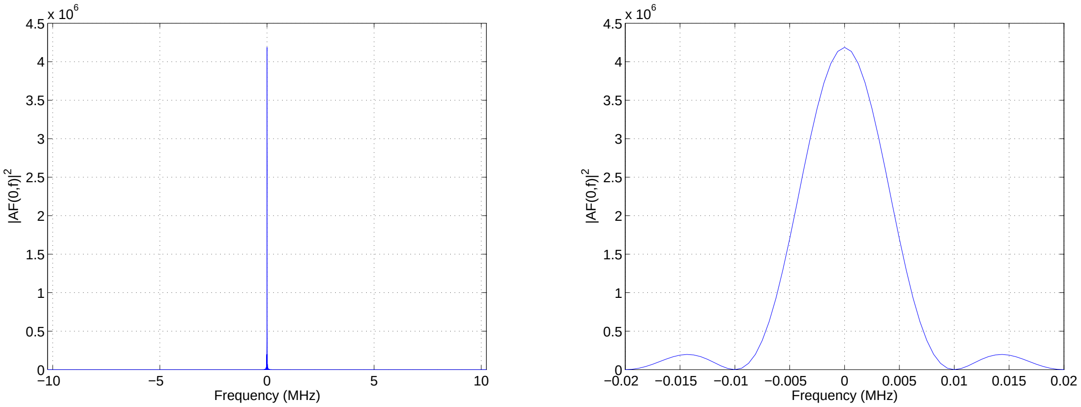

Figure 5.7: Fourier transform of the QPC matched filter response

function. As shown in Figure 5.8, we see a fairly typical auto-correlation function for compressed waveforms. That is, unlike the rectangular pulse which has a correlation envelope that falls off linearly with respect to the time delay, the correlated energy is highly grouped around the zero-lag location. This is desirable to reduce the time-delay ambiguity, and one of the motivators behind pulse-compressed signals. Finally, we look at the full ambiguity function. In Figure 5.9, we see in this zoomed plot just how localized the energy is. In Figure 5.10, we get a slightly better perspective with

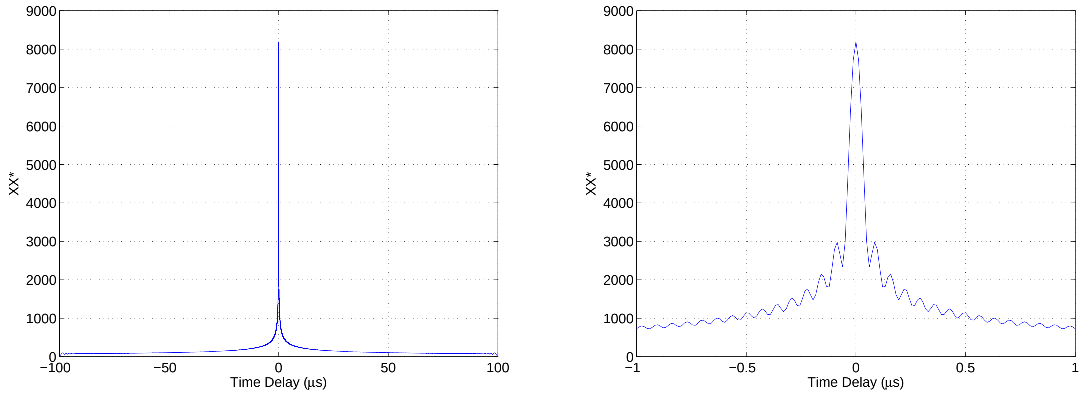

Figure 5.8: Auto-correlation function for the QPC

a three-dimensional (3D) rotated view with a red star placed on the peak for clarity. This type of ambiguity function, close to the ideal "thumb tack," is highly desirable [15]. The only disadvantage of such a distribution is that the ambiguity function must be used as the estimator, as the return is unlikely to be detected if a Doppler mismatch exists for a traditional matched filter approach. However, we have already resigned to using the peak-location of the ambiguity function as the estimator, and so this is not an issue for this specific setup. In Figure 5.11 we view the rotated 3D ambiguity function up close for a much smaller range range-rate grid. Here we can see the detail more obviously, as well as the angled cuts shown previously.

To ensure the algorithm is indeed optimizing, and that we are not just fortunate enough to have randomly selected a good waveform from the four-dimensional (4D) waveform space, we tested the baseline algorithm against a randomly generated waveform with the parameters uniformly chosen throughout the 4D space in Table 5.4. Here we see, that while it appears to perform as well as the best linear chirp in this scenario, the optimized waveform still outperforms it by at least 2 dB. This has the interesting byproduct that randomly choosing waveforms in this space is not a catastrophic method, and provides a decent PMSE and convergence rate.

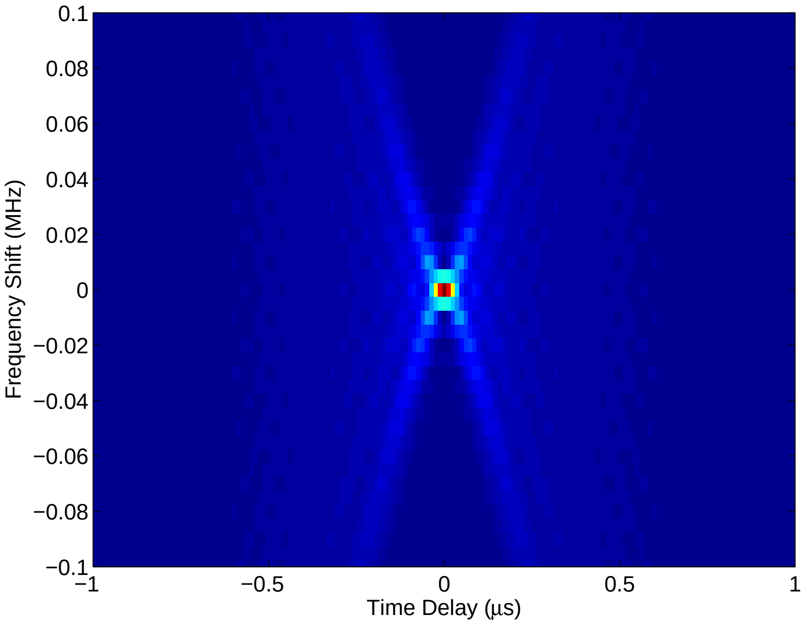

Figure 5.9: Ambiguity function of the QPC

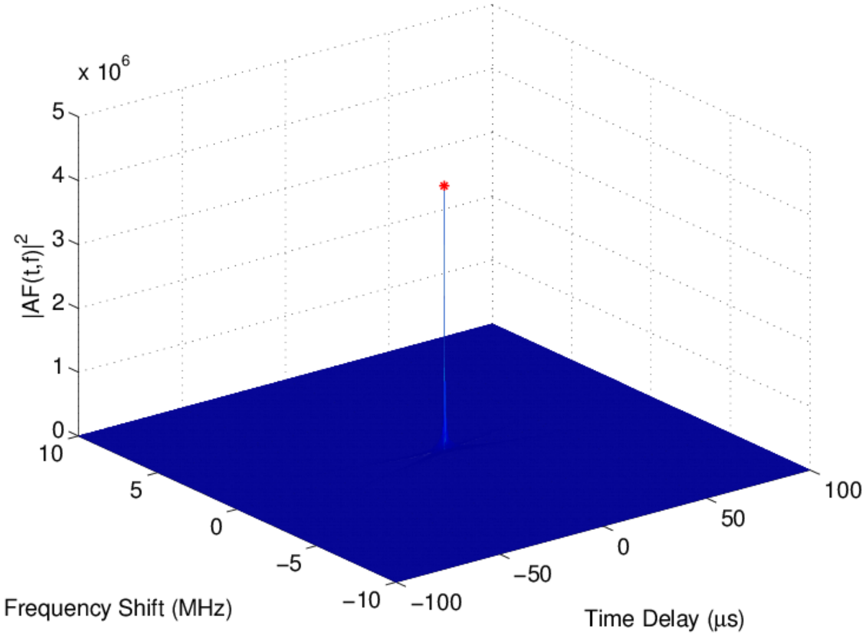

Figure 5.10: Rotated ambiguity function of the QPC

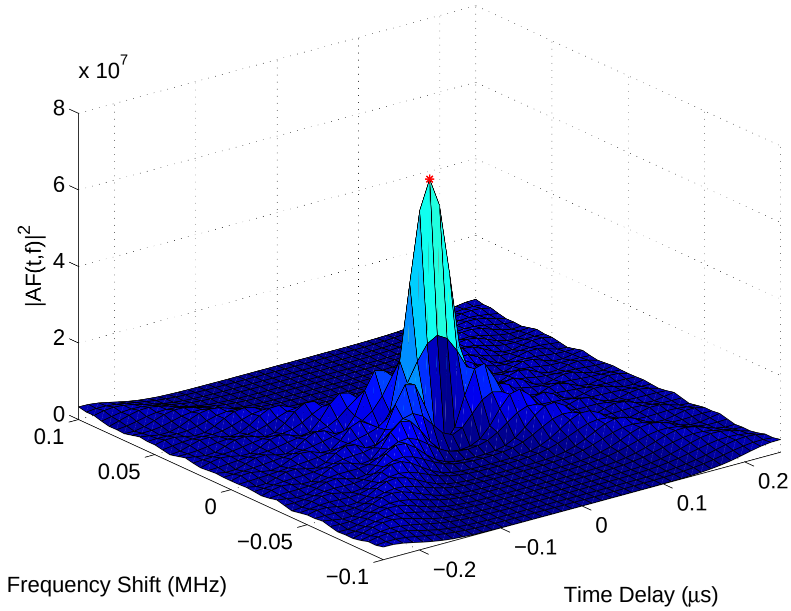

Figure 5.11: Rotated ambiguity function of the QPC, zoomed

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	39
▶ Parameters randomly and uniformly chosen	49

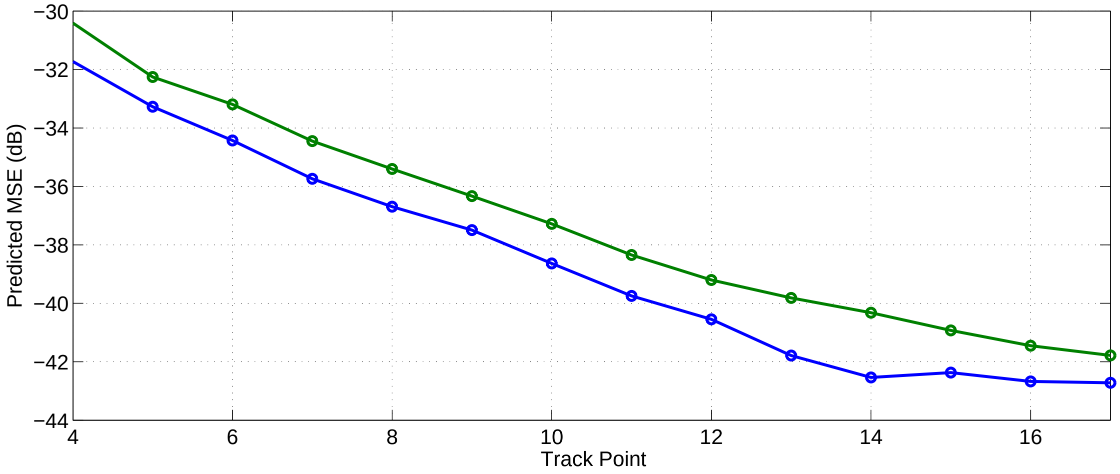

Table 5.4: Low clutter, DE vs. random waveform

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	39
▶ 25x49 = 5x5x7x7 = 1225 point 4D Grid Search	34

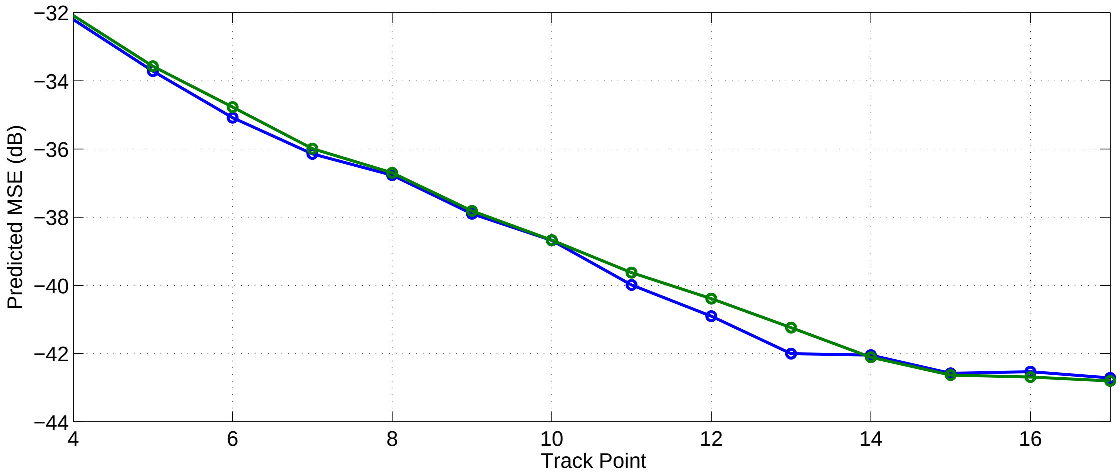

Table 5.5: Low clutter, DE vs. grid

We next wanted to demonstrate the power of these methods in contrast to grid-based methods. In Table 5.5, we look at the first of such scenarios. Here, we chose a grid with a higher number of tested waveforms than 20 iterations of the baseline DE. This is because the baseline was not factorable to an integer cube. Nevertheless, the DE method performed as well or better than the grid based method with a higher convergence rate. One thing that was noted was that the grid-method sampled the 4D space such that the extrema of the TB space were always sampled. As we have seen from previous trials, the maximum time and bandwidth appear to be heavily favored. It was at this point that the brickwall penalty was reevaluated. Initially, a brickwall penalty was imposed by setting the cost exorbitantly high for time and bandwidth parameters that fell outside of the constraints. As can be seen from previous results, the edges of the time-bandwidth space are often chosen. That is, the maximum time duration appears to be heavily favored, and in low uncertainty, the maximum bandwidth is as well. By imposing the brick-wall penalty, if candidates are proposed just outside of these limits, they are effectively discarded. This is detrimental in two ways. First, the algorithm naturally push candidates this direction since they typically have lower cost as observed by the grid-based error surface. Therefore, we must rely on the statistical probability they fall just before the edge, and not over. In addition, if the edge is the optimal point, reaching it exactly is less likely. Second, by pushing candidates outside of this region, we exclude them from that iteration and possibly subsequent iterations until they randomly mutate back inside the valid space. As a result, we function as if we had less candidates than we computationally undertake. In order to combat the brick-wall penalty deficiencies, an alternative was tested. This amounted to saturating logic, whereby candidates mutated outside of the space are saturated to their limits. While the shaping angle space continues to wrap naturally, if a candidate exceeds the time or bandwidth bounds, it was set equal to the bound. This biases the candidates toward the exact edge values, but we have seen this to be favorable, and it should only help convergence. This also keeps all of the candidates in play at every iteration and prevents the degenerate case. We tested this saturating logic on a more even ground with the grid based method in Table 5.6. Here we chose a small number of iterations and slightly more candidates in an effort to produce a factorizable product. The result is a 256 point 4D grid search so both the DE method and grid method had the same number of calls to the cost function. Here we see even more of an improvement, as much as 1 dB is some areas. However, we see that the grid based method had higher convergence in this particular case.

As proposed in [30], we have more potential to improve the rate of convergence over the grid based methods. Since the track is not expected to change rapidly with time, we can expect a relatively slowly shifting cost landscape. Indeed, if we plot the error landscape cut with respect to time, we see it vary from time-step to time-step,

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=4, N_p=64, F=0.75, Cr=0.9	37
▶ 4x64 = 4x4x4x4 = 256 point 4D Grid Search	49

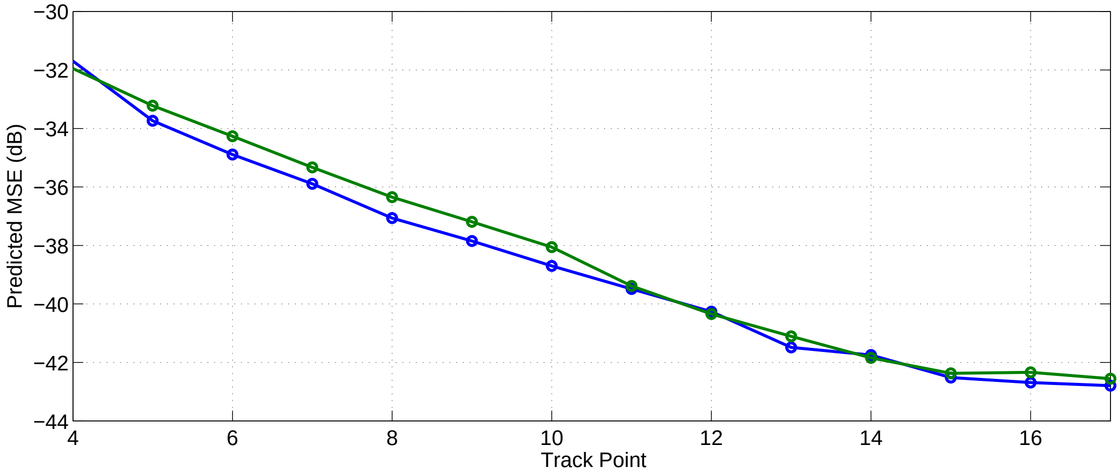

Table 5.6: Low clutter, DE vs. grid 2

but with minor changes between. Therefore, we can potentially seed the DE algorithm with the candidates from the previous iteration, but perturbed by Gaussian noise. Table 5.7 shows the results from saturating and perturbing with Gaussian noise, and we further improve over the grid-based method. Note that the cost functions must be re-evaluated after perturbation [30].

We next wanted to look at varying the knobs of the DE algorithm as we did with Beale's function and the Rosenbrock function, $n = 1$, in Chapter 4. In Table 5.8, we look at 4 modifications to the baseline algorithm. Here we see very little deviation in performance, less than a 1 dB at all points. The convergence rate appears to be well matched as well, but the baseline and increasing the number of population members seems to do the best in this regard. One thing worth noting is that since the difference between 20 iterations and 200 seems minimal, it could be that 20 iterations is too many to see any deviation in performance. That is, the number of iterations required

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=4, N_p=64, F=0.75, Cr=0.9	37
▶ 4x64 = 4x4x4x4 = 256 point 4D Grid Search	49

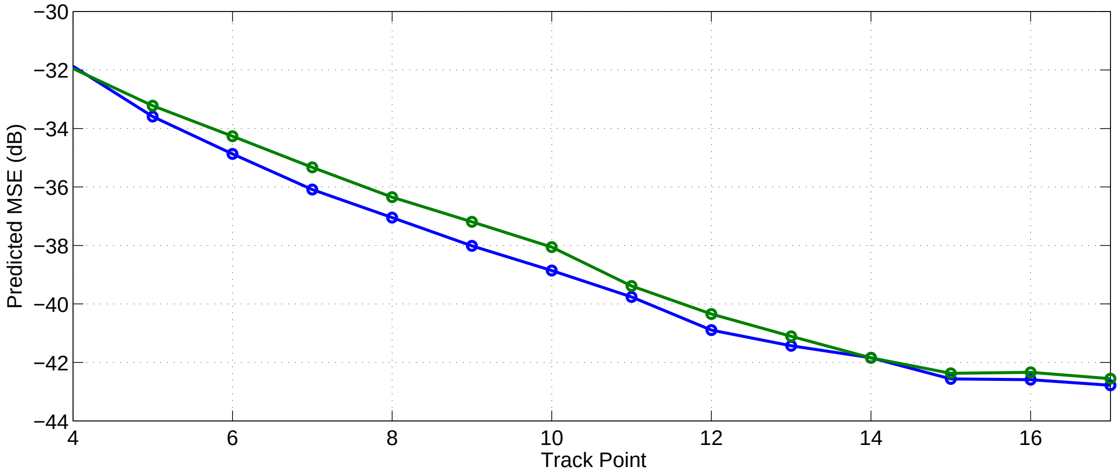

Table 5.7: Low clutter, DE vs. grid 2 with DynDE

to achieve roughly this contour of PMSE may be affected by the various settings. In Table 5.9 we compare the baseline with the remaining typical variations of DE, as explored in Chapter 4. Here we see slightly more deviation, and in fact the baseline at the most extreme spread appears to outperform some of the variants by over 1 dB. However, this is not unexpected since the baseline was chosen with the 'best' settings that tend to work in general [28]. We also see a slightly higher convergence rate amongst this group, namely the random base vector and the 2 differential terms. This seems to suggest the mutation step most heavily affects convergence. However, these two also appeared to perform the worst in regard to PMSE, albeit by a thin margin.

Since the waveform shape appeared to gravitate to the QPC, we decided to explore the tradeoffs in optimizing with respect to position only or velocity only. To start, we look at position only in Table 5.10 to compare the DE optimized for PMSE

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	39
▶ DE/best/1/bin, G_{max}=200, N_p=40, F=0.75, Cr=0.9	34
▶ DE/best/1/bin, G_{max}=20, N_p=100, F=0.75, Cr=0.9	39
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.50, Cr=0.9	37
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.5	38

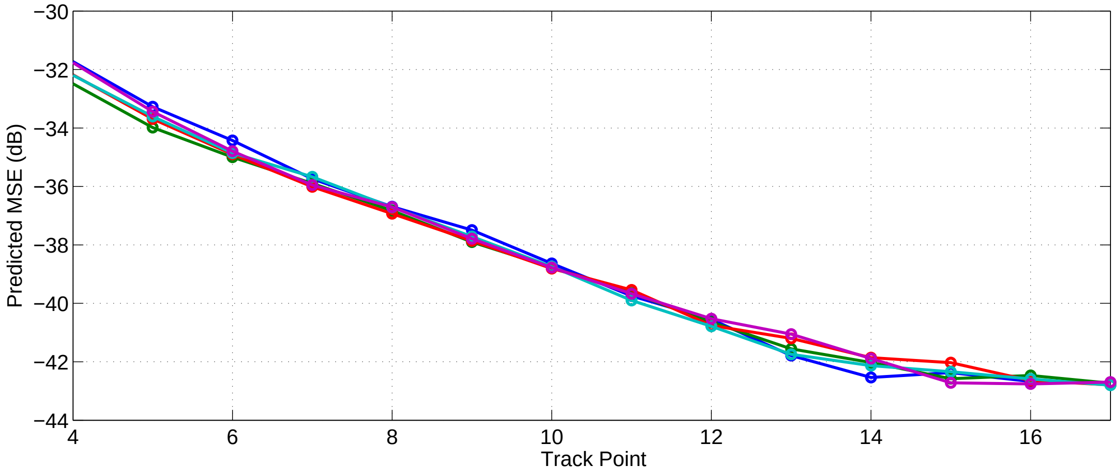

Table 5.8: Low clutter, DE vs. DE

for position only. In weighting position only, the maximum TB product bounding exponential chirp (BEC) appeared to be heavily favored instead of the QPC as shown in Figure 5.12. While again the DE method had slightly higher convergence rate, the performance margin was less than one dB. To explore why this particular waveform was favored, we again look at the various transforms. The shaping parameters ($\frac{\pi}{4}$, $\frac{3\pi}{2}$) produce "bouncing exponential" shaped instantaneous frequency shown in Figure 5.13, with polynomial coefficients $A =$ -1, $B =$ -1, $C = 0$. As we can see in Figure 5.14, the low amplitude oscillation for half of the duration puts most frequency content at 7.5 MHz, similar to the QPC. As reflected in the spectrogram, the frequency content is grouped around $BW/2$ = 7.5 MHz, as shown in the Fourier transform in Figure

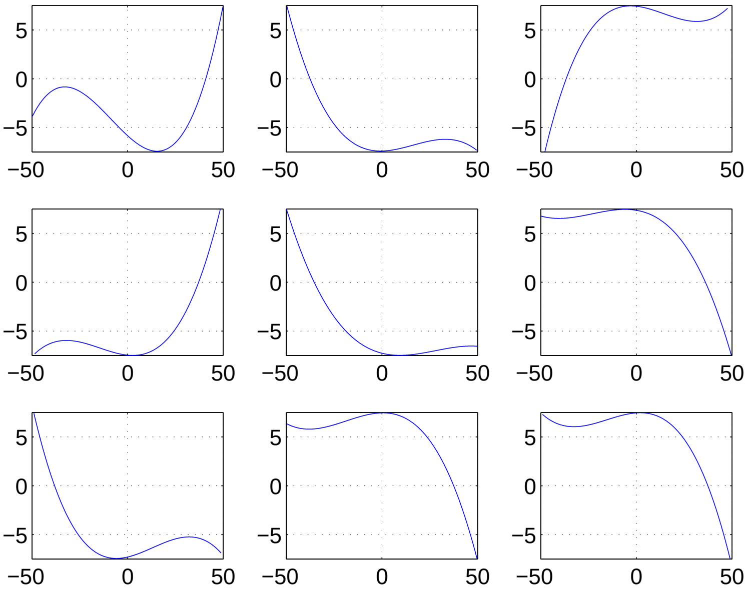

Figure 5.12: Typical IF, position only weighting frequency function for each track point (1-9)

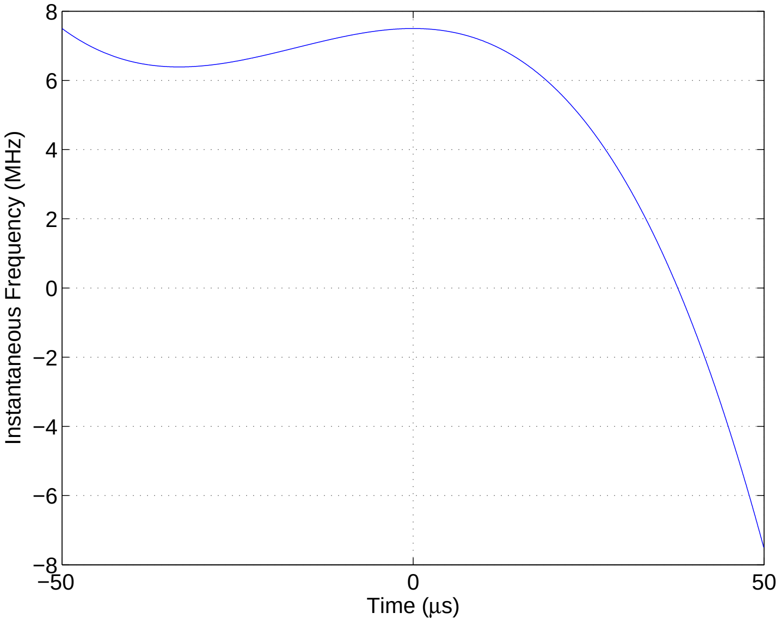

Figure 5.13: Instantaneous frequency for the BEC

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	39
▶ DE/rand/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	42
▶ DE/t-to-b/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	38
▶ DE/best/2/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	42
▶ DE/best/1/exp, G_{max}=20, N_p=40, F=0.75, Cr=0.9	39

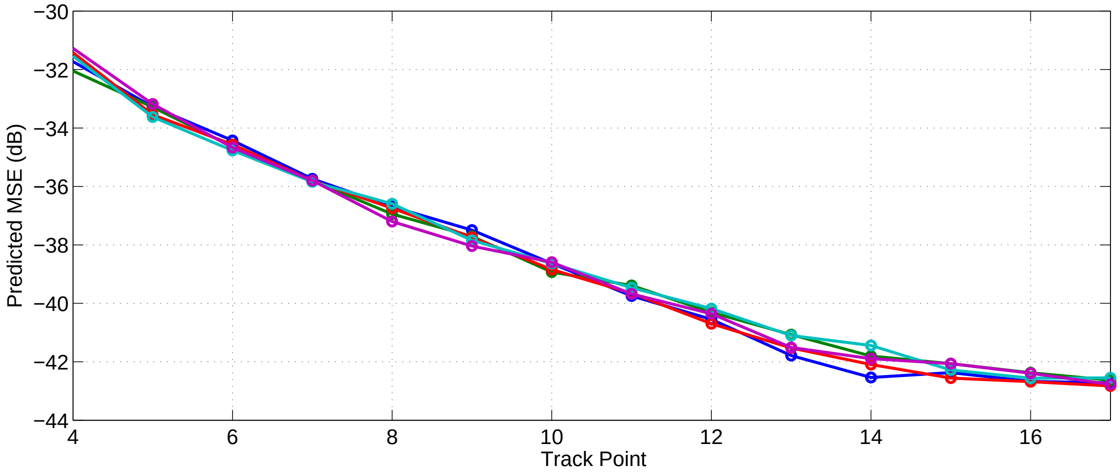

Table 5.9: Low clutter, DE vs. DE 2

5.15. This is a much tighter grouping than the QPC since it is not symmetric in frequency like the QPC is. We next look at the Fourier transform of the zero delay auto-correlation to get the zero delay cut of the ambiguity function. Here we see in Figure 5.16 that once again we get a significantly peaked spectrum. The Rayleigh Doppler resolution [2] is 10 kHz, nearly identical to the QPC. In Figure 5.17 we see the zero-Doppler cut of the ambiguity function, and similar pulse compression properties to that of the QPC. However, the sidelobe content is significantly different from the QPC. While much more energy is grouped around the main lobe, there are distinctive nulls instead of a gradual decline. Further, we see energy in bouncing exponential correlation falls off rapidly after around 60 μs. So while the main lobe is

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	37
▶ Quadratic Parabolic Chirp (BW_{max},T_{max})	36

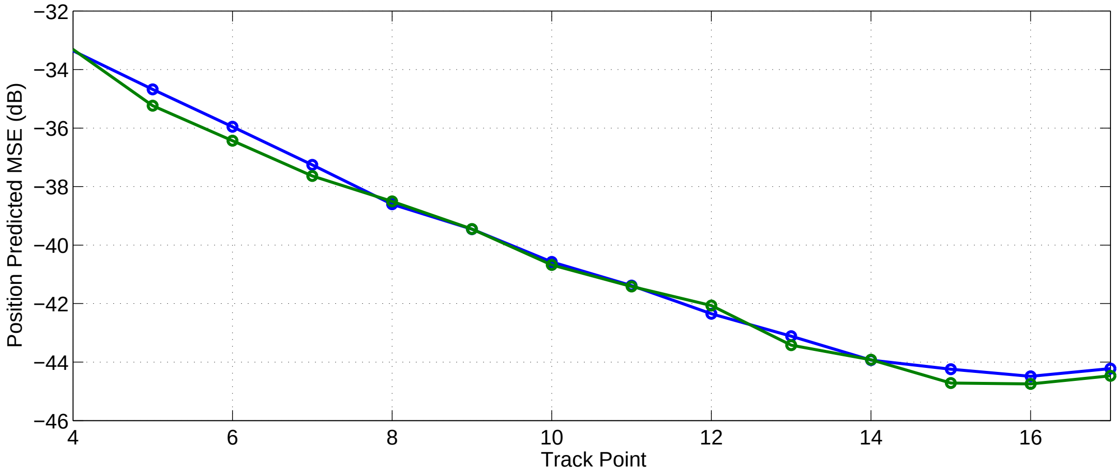

Table 5.10: Low clutter, DE vs. max TB QPC, position only

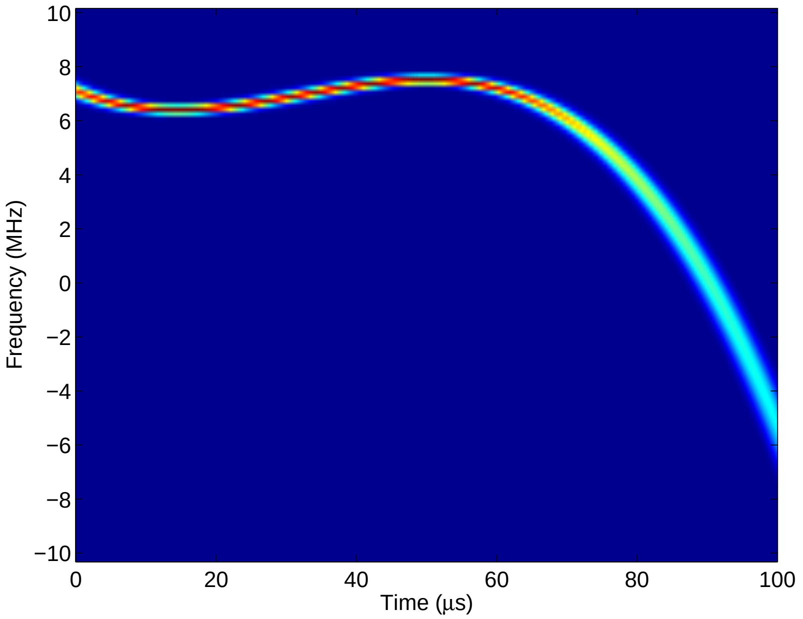

Figure 5.14: Spectrogram for the BEC

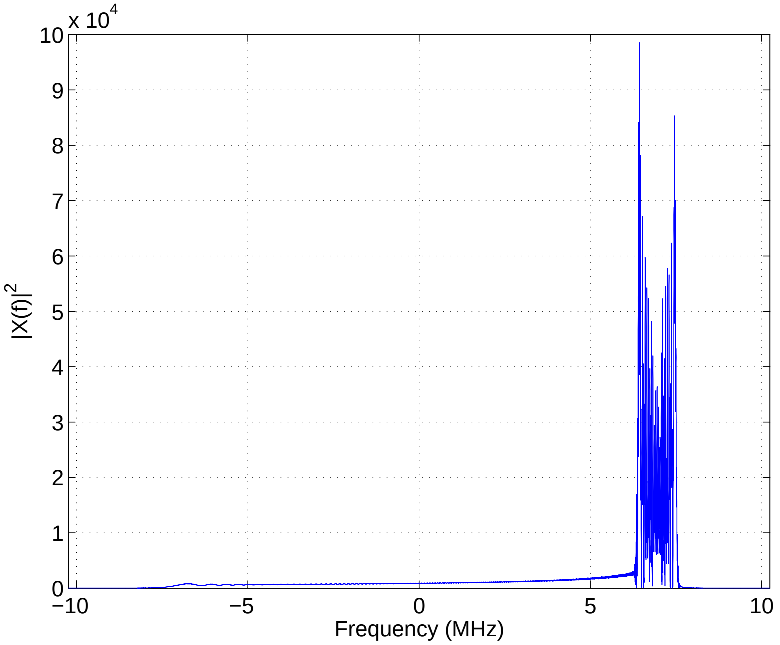

Figure 5.15: Fourier transform of the BEC

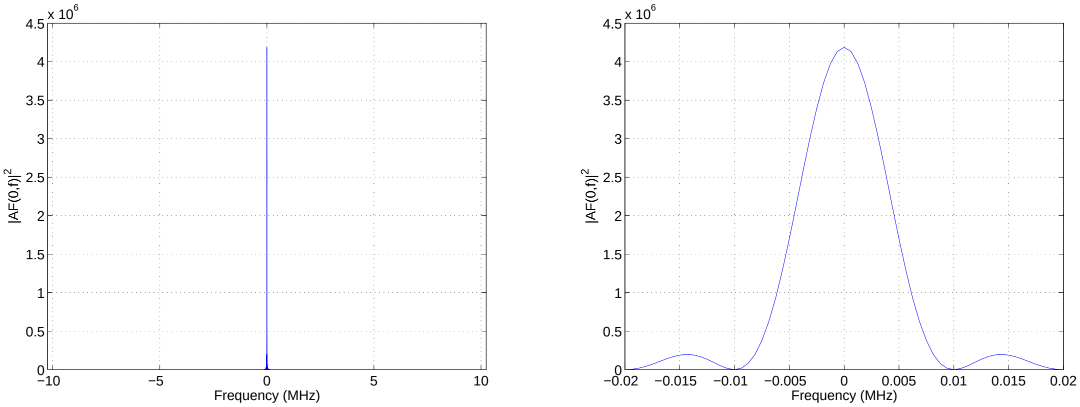

Figure 5.16: Fourier transform of the BEC matched filter response

wider, the sidelobes are faster fading and with deep nulls. To explore this trade-off in more detail, we overlay the two autocorrelation functions in the log domain in Figure 5.18. Here we see that the BEC has a 3 dB range resolution [2] of 45 ns, while the QPC has a 3 dB range resolution of 30 ns. Also, energy in the QPC does not drop below -40 dB until the end of the pulse duration. Since the main lobe is only characterized in the system, it is difficult to gauge the relative time performance here, but it is interesting that this particular frequency-modulated (FM) waveform

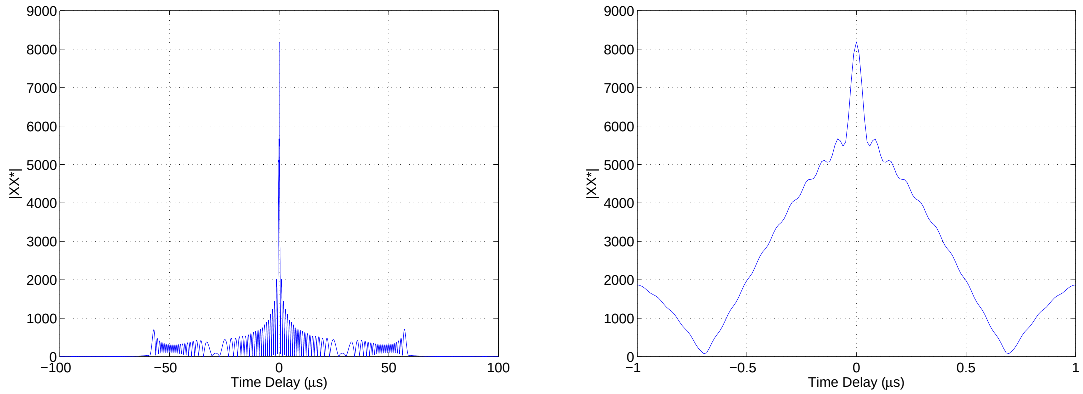

Figure 5.17: Auto-correlation function for the BEC

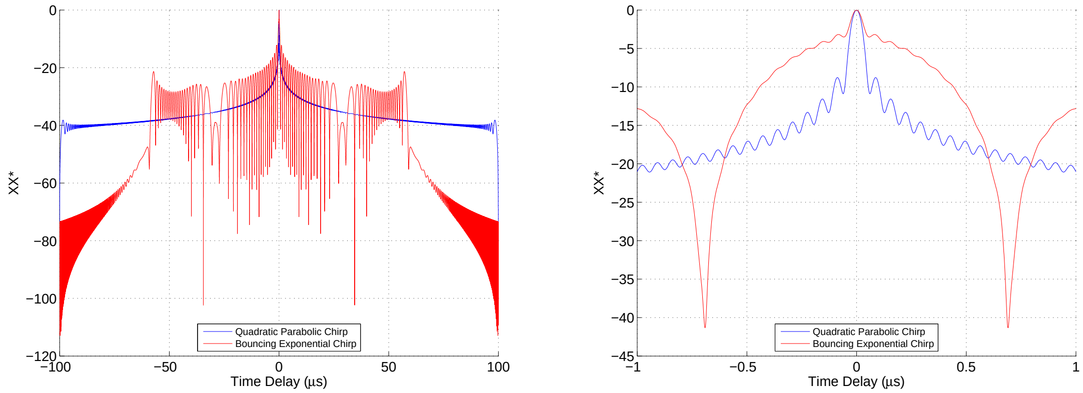

Figure 5.18: Comparison of auto-correlation of the QPC and the BEC

was chosen when position was weighted instead of velocity. Here we again see in Figure 5.19 the energy still highly localized in range-rate dimension. While some smearing in the range dimension is present from the autocorrelation shape seen on the previous plot, it is still relatively concentrated over the whole Doppler plane. Figure 5.20 is more telling, and in this dimension we can see the significiantly higher sidelobe content.

We then looked at weighting velocity only, and in doing so, found that a reduced bandwidth QPC was favored during acquisition (the first few points of the track) as shown in Figure 5.21, but otherwise the standard QPC was favored. This has some interesting implications. First, it suggests that the velocity dominates the error function for this particular track. That is not surprising since the time-varying velocity

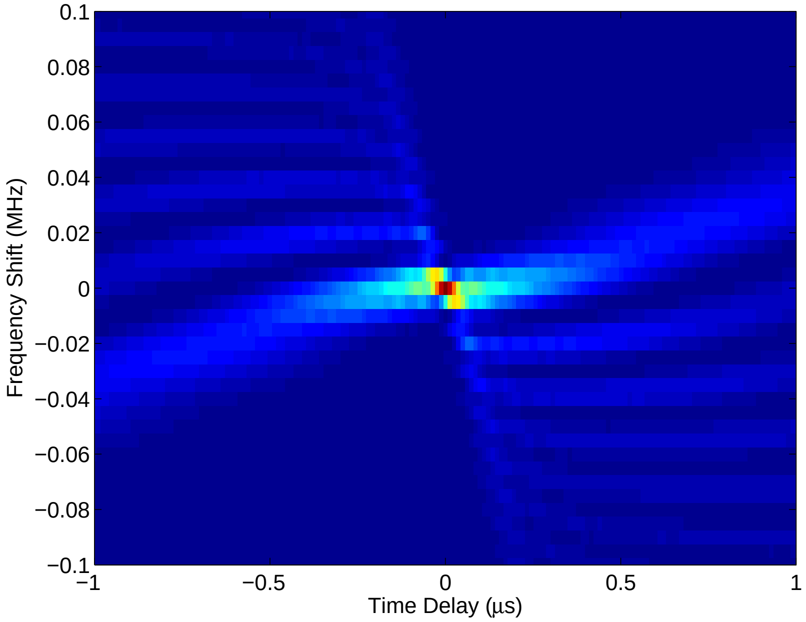

Figure 5.19: Ambiguity function of the BEC

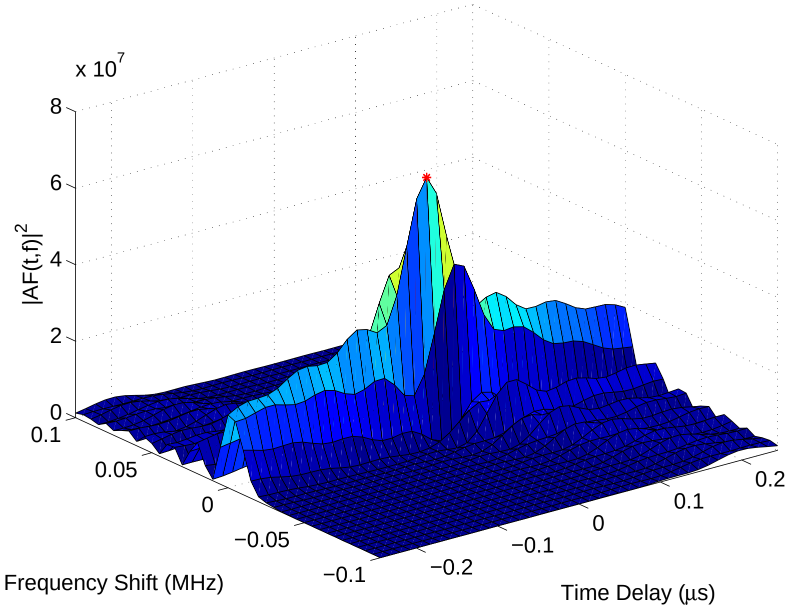

Figure 5.20: Rotated ambiguity function of the BEC, zoomed

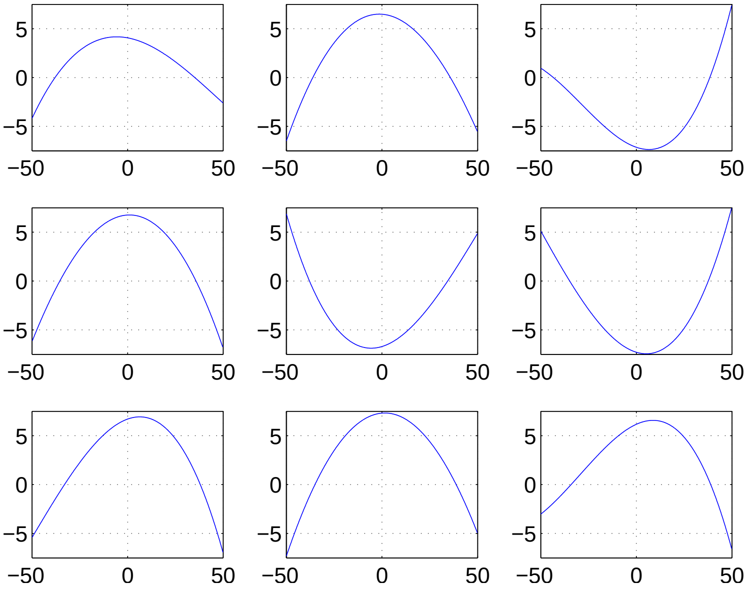

Figure 5.21: Typical IF, velocity only weighting frequency function for each track point (1-9)

violates the underlying motion model assumption. Second, we see that during the time of most uncertainty, when the track is acquiring, the bandwidth is reduced. This results in a larger validation gate region [3], and thus allows for a greater likelihood that the detection is within the validation gate. Sure enough, we see an increased margin the convergence rate in Table 5.11. While the PMSE margin is still thin, less than one dB, the modulation of the BW appears to help convergence rate when considering the velocity only.

In order to provoke a more interesting waveform selection out of the optimization algorithm, the clutter density ρ was increased by an order of magnitude to 10^{-4}. This had many interesting consequences. First, we note the linear chirp results in Table 5.12. The upswept and downswept chirps were combined into a single trial result because these two particular fixed waveforms were the only two fixed linear chirps to converge. The remaining time-bandwidth points (upswept and downswept) did not produce even a single convergent track within 100,000 trials. Another interesting

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	39
▶ Quadratic Parabolic Chirp (BW_{max},T_{max})	36

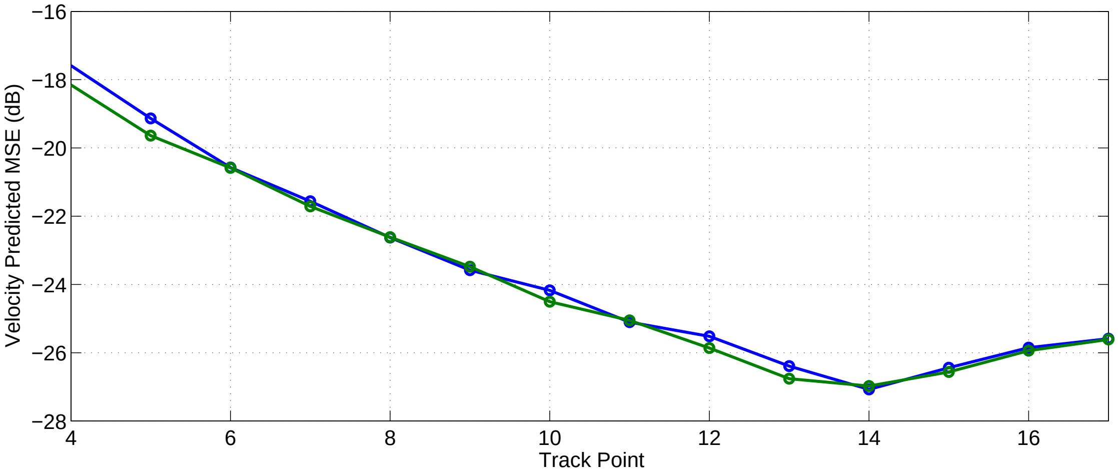

Table 5.11: Low clutter, DE vs. max TB QPC, velocity only

thing to note is that the worst performing linear chirps from the low clutter scenario are the only two tracks left, and the best performing linear chirp, the maximum TB product, is notably absent. However, these were the two tracks that had the highest convergence. It appears that convergence rate and PMSE are opposing requirements, but yet we now see the gap in convergence rate completely closed, as the DE trial outperformed the linear chirp methods by 5 dB with a convergence rate as high as the best linear chirp.

To explore this notion, we look at a typical set of high clutter configured waveforms in Figure 5.22. Much like the velocity only weight in low clutter, we see a distinct BW modulation that appears to help acquisition (and thus convergence rate). Therefore it appears the high BW waveforms, while good for PMSE, narrowed the validation gate too much and performed very poorly in the face of high clutter. Interestingly, we still see no modulation in the pulse duration, with nearly all chosen waveforms

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	12
▶ Upswept Linear Chirp (BW_{min},T_{max})	12
▶ Downswept Linear Chirp (BW_{min},T_{max})	6

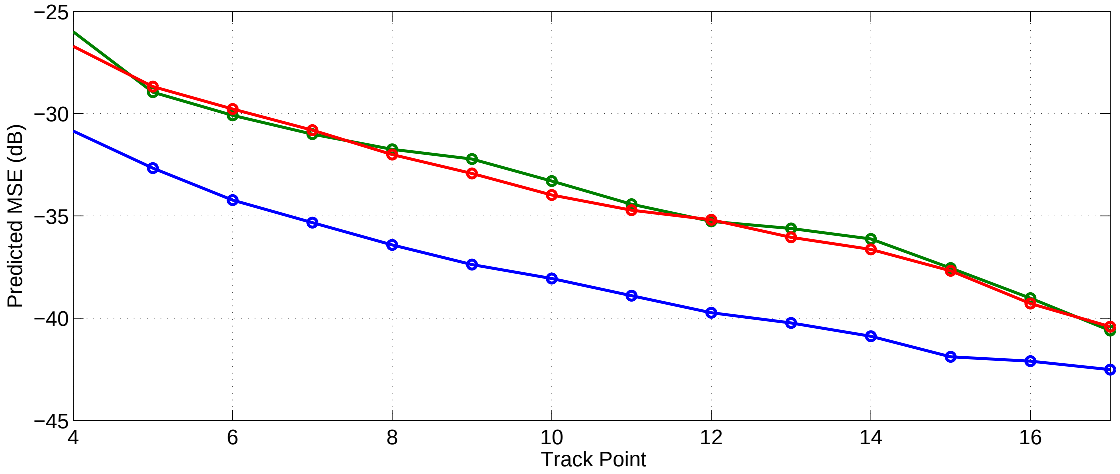

Table 5.12: High clutter, DE vs. linear chirps

using the full pulse duration available.

In Table 5.13, we compare the high clutter scenario baseline DE algorithm with the QPC. As before, we have only two surviving fixed waveforms, but this time the maximum TB product QPC still converged and performed as well as the optimized trial as before. However, once again, convergence is higher for the DE solution.

To again compare with a randomly chosen waveform for the sake of completeness, we look at Table 5.14. Here we see another interesting result. First, the gap in performance has widened. It appears with high clutter, randomly choosing a waveform is no longer a suitable strategy. Indeed, the performance gap is now as high as 5 dB. Further, the convergence rate, which was higher in the low clutter case, is now lower than the configured waveform. It appears that with higher clutter, the performance disparity of the DE configured system and competing methods widens across

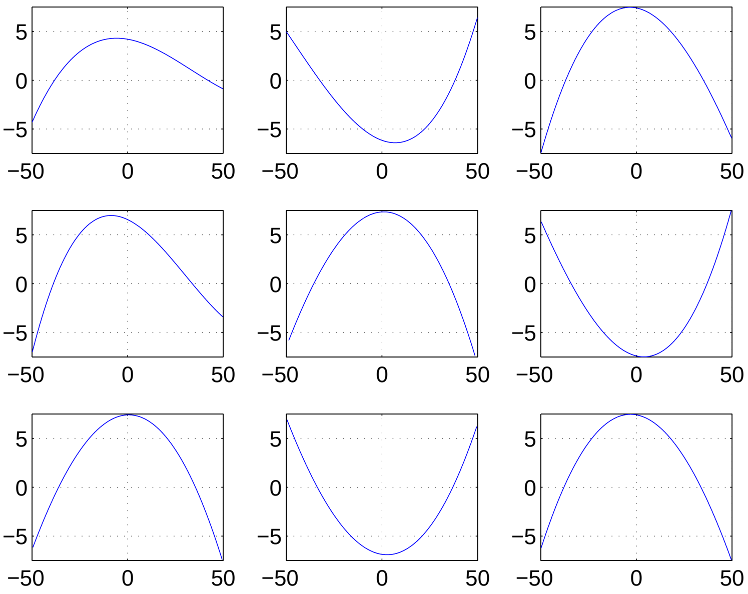

Figure 5.22: Typical IF, high clutter instantaneous frequency for each track point (1-9)

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	12
▶ Quadratic Parabolic Chirp (BW_{min},T_{max})	10
▶ Quadratic Parabolic Chirp (BW_{max},T_{max})	10

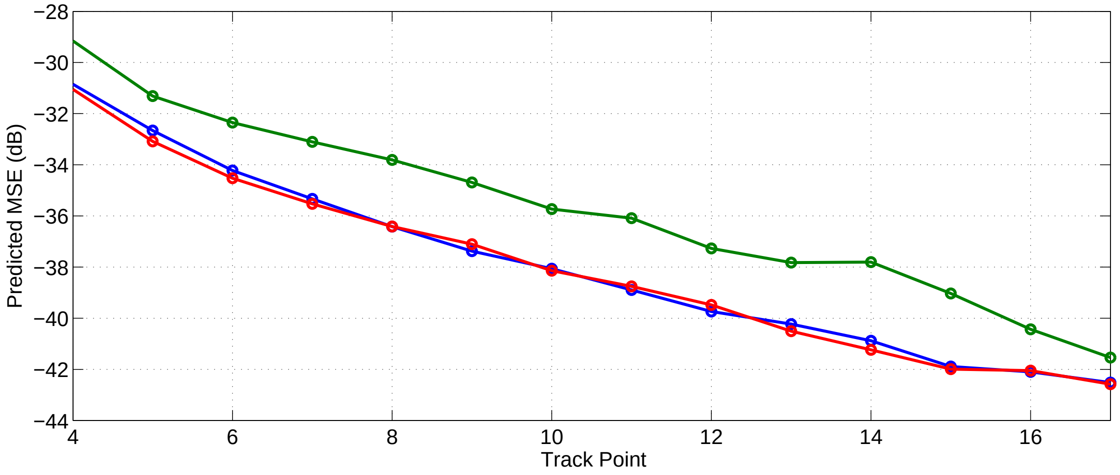

Table 5.13: High clutter, DE vs. QPCs

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	12
▶ Parameters randomly and uniformly chosen	7

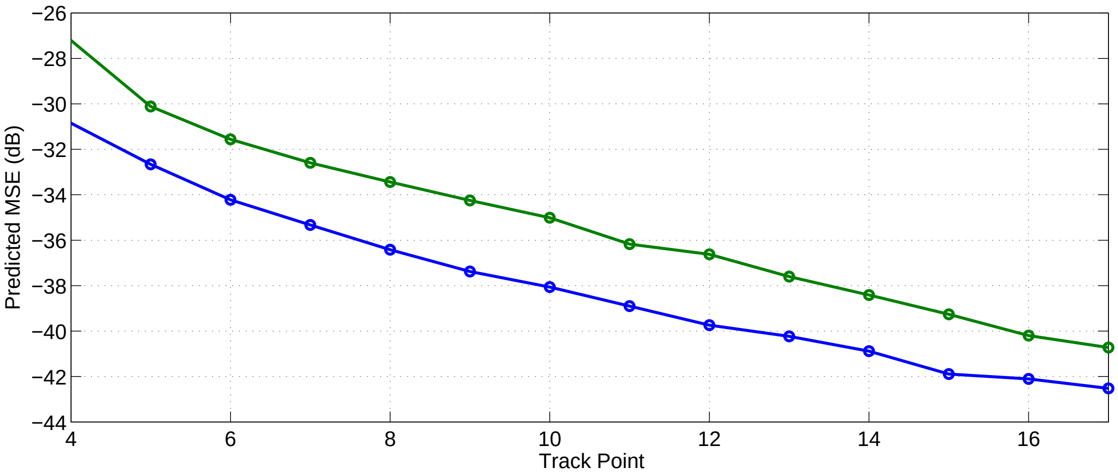

Table 5.14: High clutter, DE vs. random waveform

the board, with exception to the QPC and PMSE.

Since the velocity error appeared to dominate the optimization process, we wanted to look at a new scenario where the velocity was indeed constant. As a result, there is no acceleration, but still an initial uncertainty as in the previous case. The constant velocity track of choice is shown in Figure 5.23. In this case, we placed the sensor at (20000, 2500), near the tangential center of the track. As before, we chose a distance with a high SNR for $P_D > 0.95$, but not so high to create a peaked likelihood and poor particle filter performance [31]. The only non-constant velocity portion of this track arises from the initial acquisition which is dispersed with the same covariance as the previous track. Therefore, the noise intensity was reduced to $q_k = 100$.

The first test shown in Table 5.15 again compares DE against the upswept linear chirps with the extrema of the TB space. A multitude of observations can be immediately drawn. First we note that aside from the first point, the minimum time

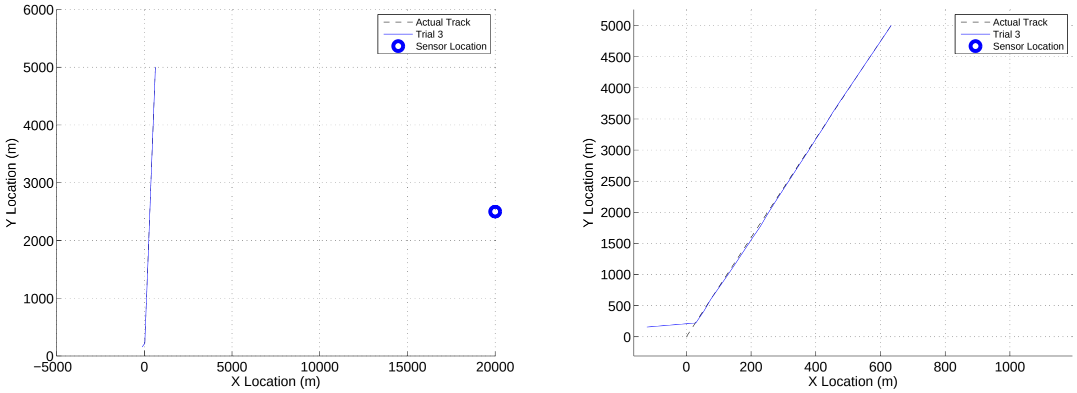

Figure 5.23: Constant velocity track with example simulation overlay

Test Description	**Convergence Rate**
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	79
▶ Upswept Linear Chirp (BW_{min},T_{min})	82
▶ Upswept Linear Chirp (BW_{max},T_{min})	34
▶ Upswept Linear Chirp (BW_{min},T_{max})	91
▶ Upswept Linear Chirp (BW_{max},T_{max})	53

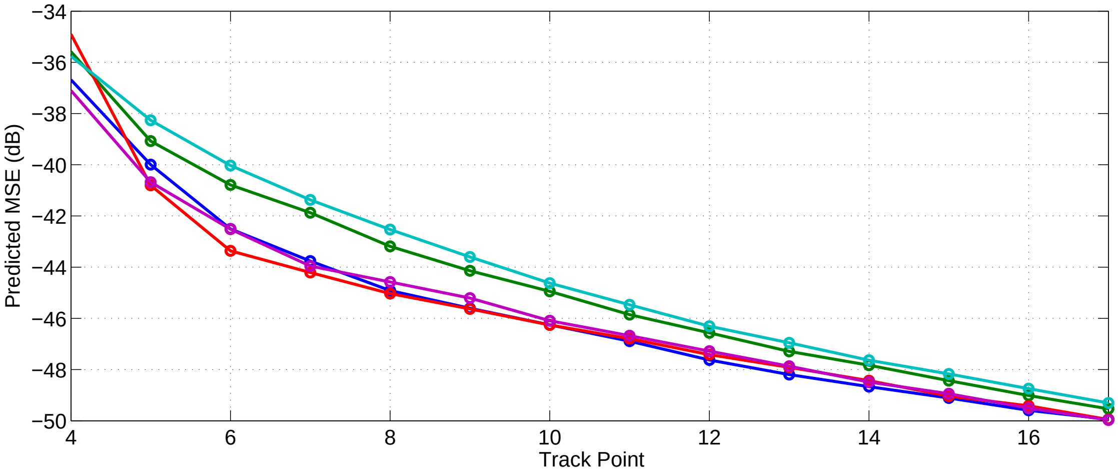

Table 5.15: Constant velocity, DE vs. upswept linear chirps

duration, maximum bandwidth linear chirp performs as well as the maximum TB linear chirp. In the previous track, anything other than the maximum time duration was never favored. In addition, these two signals have a similar performance to the configured DE baseline. In the previous track, the gap was 5 dB at some points, and a minimum of 1 dB at all others. While the minimum bandwidth, maximum time duration linear chirp still has the highest convergence rate, the gap in convergence rate has narrowed over the configured waveform. Interestingly, the two signals that performed as well as the dynamic waveform had the worst convergence rate by a wide margin. While in the previous track we noticed a more clear trade between PMSE and convergence, here we see the DE optimized waveform does as well as the best linear chirp with a high convergence rate.

For the downswept linear chirps, shown in Table 5.16, we see very similar results. This time, as before, the suite of downswept linear chirps had slightly lower convergence rate than their respective upswept chirps. As a result, the convergence rates are beat by the DE in all but one case. To gain insight into how the DE is optimizing with respect to this new track, we once again sample some track points to see what typically is chosen for the IF functions. In Figure 5.24 we see a much more rich set of chosen functions. To start, the first point is close to the minimum time duration, and constrained below the maximum bandwidth. The next couple points look more like waveforms we experienced with the first, turning track, but with less than the maximum bandwidth. Finally, a series of waveforms ranging from some close to the QPC and the BEC are chosen after acquisition. Here we clearly see the benefit to waveform design; when the target dynamics changed with respect to the sensor location, the optimal waveform has changed. Where in the previous track less than the maximum time duration was never selected, here it is competing for one of the best waveforms.

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	79
▶ Downswept Linear Chirp (BW_{min},T_{min})	77
▶ Downswept Linear Chirp (BW_{max},T_{min})	32
▶ Downswept Linear Chirp (BW_{min},T_{max})	81
▶ Downswept Linear Chirp (BW_{max},T_{max})	48

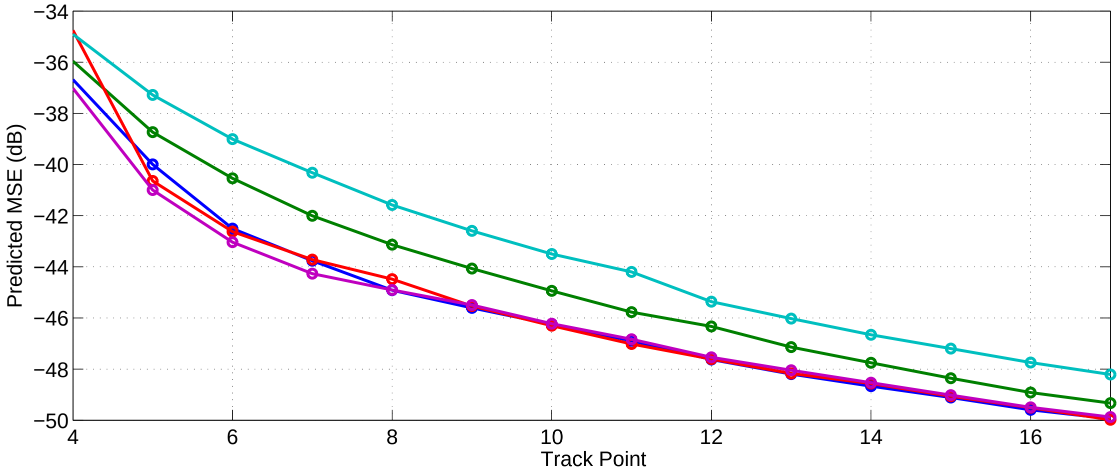

Table 5.16: Constant velocity, DE vs. downswept linear chirps

Table 5.17 shows how we perform relative to the QPC, the best waveform in the previous track. Here we see an interesting trend. First, the maximum TB QPC no longer performs the best. In fact, it suffers during the first half of the track. The adaptive DE solution appears to start with a small pulse duration perhaps to help with acquisition as the frequency modulation did in the first track. After acquisition, the time duration is maximized again since the uncertainty is removed. Further, the convergence rate is much higher for the DE than the best QPC.

Since a diverse waveform set was chosen, the random waveform test is of even greater interest. In Table 5.18, we see that the performance margin between the random and the optimized has closed dramatically. There is a gap at some points

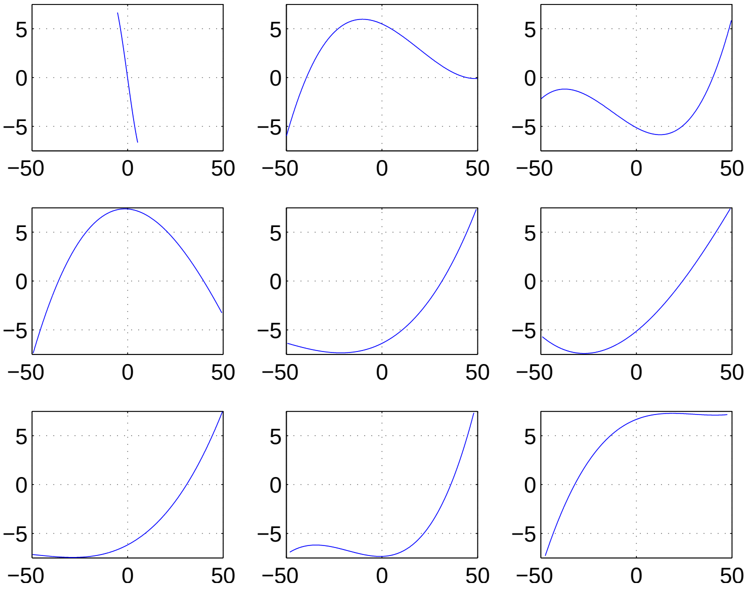

Figure 5.24: Typical IF, constant velocity frequency function for each track point (1-9)

as large as 1 dB, but other than that the margin is thin. This suggests that with a proper motion model, there are many more “good” waveforms than bad, though we have seen from some of the previous plots picking a bad one as a fixed waveform suffers significant degradation in performance, as much as 4 dB for this track. Of course, if the motion model is known precisely *a priori*, then there would be no need to process the track dynamically. Only the original point would need to be estimated with this being refined with time as more measurements are made to converge on the state. Therefore, while the random waveform appears to have some interesting implications, it is not optimal. We also note that the DE had a higher convergence rate, in contrast to the other track.

As before, we wanted to investigate the position only and velocity only weighting for this new track. Table 5.19 shows the position only weighting. This time, since we had two fixed waveforms that performed nearly as well, we included both the best downswept linear chirp and QPC. One thing that stands out is that once

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	79
▶ Quadratic Parabolic Chirp (BW_{min},T_{min})	81
▶ Quadratic Parabolic Chirp (BW_{max},T_{min})	53
▶ Quadratic Parabolic Chirp (BW_{min},T_{max})	92
▶ Quadratic Parabolic Chirp (BW_{max},T_{max})	75

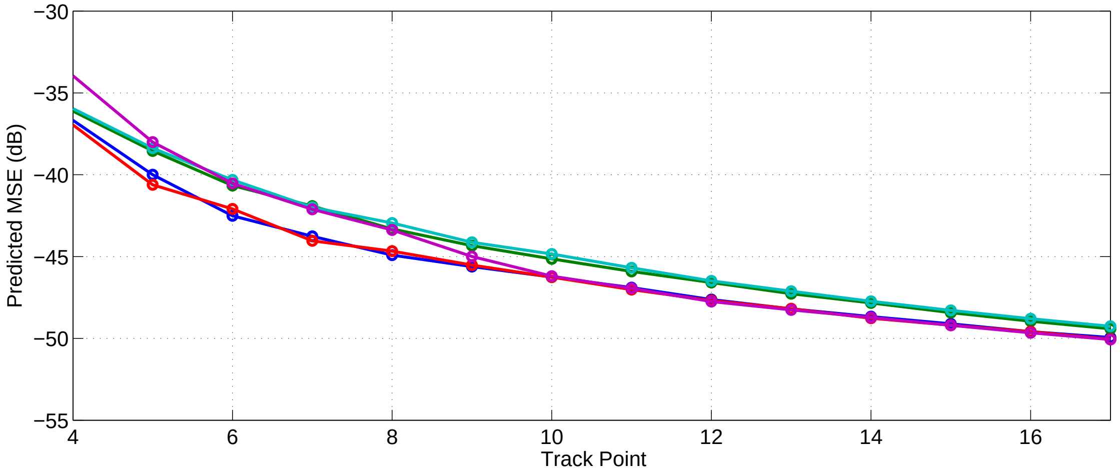

Table 5.17: Constant velocity, DE vs. QPCs

again, optimizing with respect to position only does not seem to help the configured waveform do any better than the best fixed waveform. In fact, the performance got slightly worse, as much as 3 dB in some points. Since the initial point is unknown, and since we have decreased the noise intensity q_k designed to help push probability off the constant velocity track, it could be that optimizing for position exacerbated this uncertainty. While the position motion model is designed to handle time varying position, the velocity is not, and so the acquisition phase could have taken longer as evidenced by the plot shown. After the 9th track point, the performance margin is gone.

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	79
▶ Parameters randomly and uniformly chosen	76

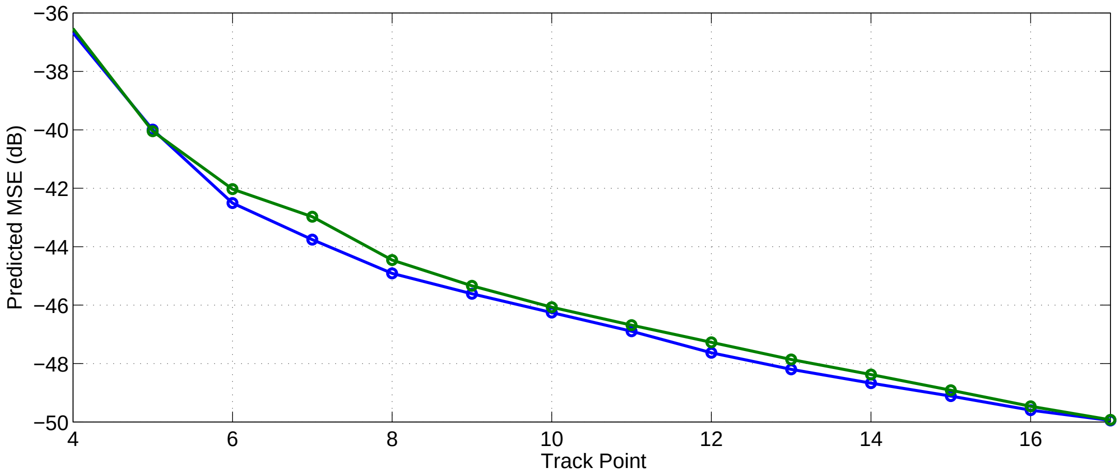

Table 5.18: Constant velocity, DE vs. random waveform

For velocity only weighting, Table 5.20 sheds some more light on this phenomenon. Here the performance margin is extremely thin, 0.5 dB at most. This emphasizes the previous point, that by weighting by position only, the PMSE was dominated by the velocity error. This is because a velocity error causes the predicted position in the Bayesian model to be incorrect. Since q_k is smaller for this track, the model is weighted more heavily in the likelihood [11]. As a result, the acquisition phase dominates the position error, but the optimizer is looking only to decrease the position error directly, not as it is affected by a high variance velocity estimate.

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	78
▶ Downswept Linear Chirp (BW_{max},T_{max})	48
▶ Quadratic Parabolic Chirp (BW_{max},T_{min})	53

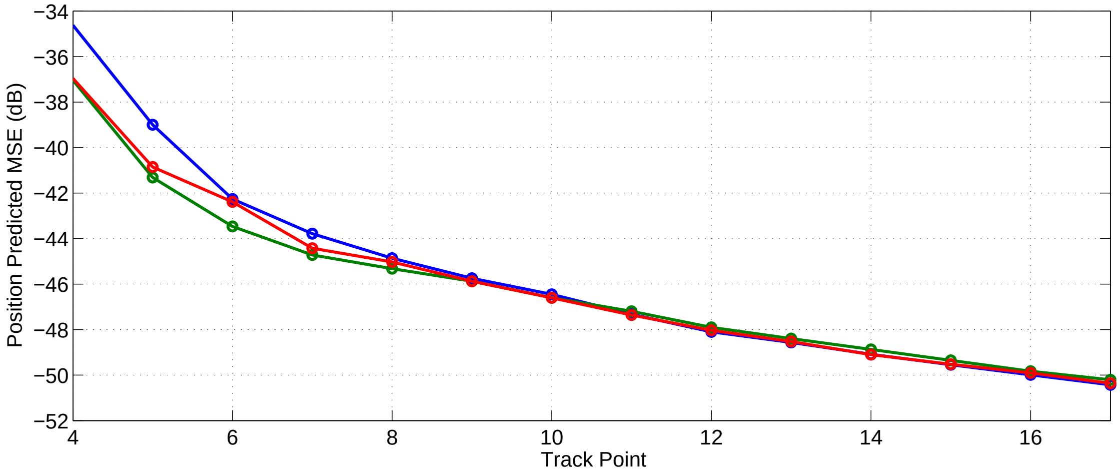

Table 5.19: Constant velocity, DE vs. best fixed chirps, position only

Test Description	Convergence Rate
▶ DE/best/1/bin, G_{max}=20, N_p=40, F=0.75, Cr=0.9	84
▶ Downswept Linear Chirp (BW_{max},T_{max})	48
▶ Quadratic Parabolic Chirp (BW_{max},T_{min})	53

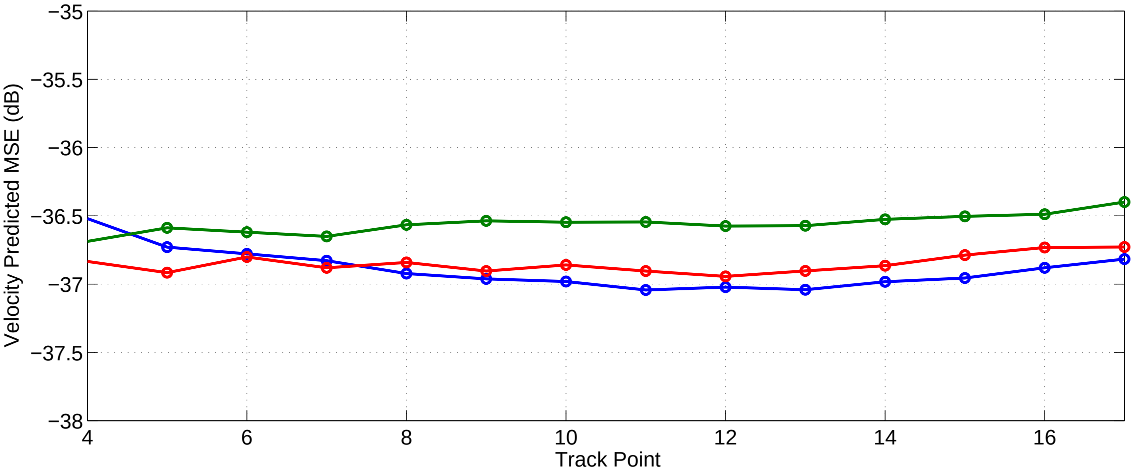

Table 5.20: Constant velocity, DE vs. best fixed waveforms, velocity only

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Summary and Concluding Remarks

In this thesis, we proposed a novel parameterizable frequency-modulated (FM) waveform with a quartic polynomial phase function (QPPF), the cubic polynomial FM (CP-FM) waveform. We started by defining the system model and two-dimensional (2D) tracking geometry, and introduced the bootstrap particle filter (BFP) to perform Bayesian state estimation based on noisy nonlinear measurements from the Doppler processor, the ambiguity function. We then motivated and derived the QPPF waveform, and showed efficient normalization that reduced the waveform parameter space dimensionality and separated shaping parameters from time-bandwidth parameters. We went on to review differential evolution (DE) and the advantages of genetic methods for complicated, non-convex optimization problems, like nonlinear tracking in clutter. In the preceding section, we saw many different results across the various tracks, scenarios, and variations. Here we distill the most important results from the various trials and make concluding remarks.

Optimal waveform shape can change depending on the scenario, motivating the shaping parameters

As we saw in the difference between the first and second track, the optimal waveform shape changed from the quadratic parabolic chirp (QPC) to the downswept linear chirp, or the exponential FM signal commonly selected. This suggests that in a more complicated track, or perhaps if we concatenated the two tracks, that the DE optimization would adapt its shape more frequently.

Optimal time-bandwidth parameters can change depending on the scenario, motivating the TB parameters

In the first track, the maximum time-bandwidth product was selected most of the time. In the second track however, the maximum time-bandwidth QPC performed poorly. Further, the higher time-bandwidth (TB) waveforms converged less often in high clutter for the first track as well. This suggests that the optimal pulse duration and bandwidth varies from scenario to scenario.

DE works as well or better than grid methods with less cost function calls

As we saw in the baseline turning track, the DE methods outperformed or performed as well as all of the grid based methods that contained more cost function calls or as many. Further, we found many more options that could be employed in the DE optimizer to improve convergence with less cost function calls such as saturating logic and DynDE [30].

DE allows for search of waveforms within a continuous shape and TB space

Waveforms that may not be captured within a finite grid were chosen using DE since it is continuous by its very nature. As we saw in some of the frequency function samples in the various scenarios, very subtle variation in the selected waveform is possible and often employed to track subtle changes in the cost landscape.

Optimized waveforms perform as well as the best fixed waveform for

a given scenario, but with higher convergence rate

Frequently we saw that even when we found what the best waveform was, and fed that waveform back in as a fixed waveform, the DE dynamic waveform selection performed as well but with a higher rate of convergence. Often we saw that while the shaping parameters may have matched the best fixed waveform, the TB parameters modulated in response to uncertainty and high clutter to secure a higher number of convergent tracks.

Optimizing using DE outperforms random sampling of the waveform space

The randomly selected waveforms did not catastrophically fail, suggesting the waveform space contains more valuable waveforms than not, a goal we set when deriving the CP-FM waveform in Chapter 3. Nevertheless, we saw a margin between the optimized waveform and a randomly selected one, suggesting the optimized waveform is intelligently selecting the best waveform, even from a good set to begin with. While the margin was smaller in the constant velocity track, this is largely attributed to the proper motion model, something not guaranteed to be found in practice. In the case of uncertainty, as in the turning track, the margin was substantial.

6.2 Future Work

There is potential for significant improvement of the accuracy of the predicted mean-squared error (MSE), and so re-evaluating the comparison of DE to grid-based methods for more complicated cost functions would be an interesting extension to the work here. In addition, the clutter model is perhaps overly pessimistic in general, as typical scenarios can discriminate clutter in range-rate, or spatially/temporally using space-time adaptive processing (STAP) [2].

As explored heavily in the DE literature, time-varying mutation and exchange parameters could be explored to decrease convergence time, and ultimately cost.

One possible method to constrain the polynomials is to not parameterize the cubic coefficients, but perhaps use a set of basis functions composed of cubic polynomials to enforce monotonicity if desired. For example, cubic Hermite splines with specific parameters (Fritsch-Carlson method) or orthogonal polynomials such as Legendre polynomials may lead to reduced optimization spaces as well.

Investigating which waveforms were frequently rejected might lead to a portion of the waveform-space that has undesirable properties in most situations. This could be used to further constrain the optimization process to decrease convergence time.

One of the main motivators given for the third order polynomial instantaneous frequency (IF) function was to avoid oscillatory chirps. Indeed, there seems to be an interest in non-monotonic IF functions, and so investigating fourth order polynomial IF spaces or higher may yield some interesting waveforms with better ambiguity properties. In this higher order space, we might see some more diverse modulation of the shaping parameter set.

In addition, incorporating narrowband interferers into the cost function would allow for selection of waveforms to suppress these interferers, such as those proposed in [25]. In regard to nonlinear IF functions, extending the results to wideband environments would have interesting implications since this space tends to be dominated by nonlinear IF functions.

In addition, these methods could be extended for more basic matched filter design. That is, depending on the environment, the returned waveform could be significantly affected by the more general Doppler scaling, and as result be significantly mismatched to the ideal matched filter. These optimization methods could be used to predict the mismatch and propose pre-distorted waveforms that would distort in the environment to match more highly with the fixed, zero-Doppler matched filter. This would have obvious benefits in receiver complexity.

In that vein, wideband environments may subsequently favor different waveforms than the ones selected here. It would be interesting to see the time-frequency properties of the waveform selected, if different. Even if it selects similar waveforms, the wideband ambiguity function of those same waveforms would shed some light on the properties of the waveform in that environment.

Finally, using the complete ambiguity function as has been done in previous efforts, even generalized to track-before-detect scenarios, could allow for much more complicated cost functions. For example, narrowband interferers could be hypothetically measured between pulse transmits and inserted into the cross-ambiguity function and the resulting output could be used to compute a cost designed to suppress the interferers.

All in all, the proposed method provides an extension to previous research, and lends itself to more complicated waveform spaces and cost functions that previous methods would fall short. It has demonstrated a clear performance margin over fixed waveforms in various scenarios, and is highly extendable to explore many future potential scenarios.

REFERENCES

[1] D. J. Kershaw and R. J. Evans, “A contribution to performance prediction for probabilistic data association tracking filters,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 3, pp. 1143–1148, January 1996.

[2] M. A. Richards, *Principles of Modern Radar: Basic Principles*, J. A. Sheer and W. A. Holm, Eds. Raleigh, North Carolina: SciTech Publishing, 2010.

[3] S. P. Sira, A. Papandreou-Suppappola, and D. Morrell, “Dynamic configuration of time-varying waveforms for agile sensing and tracking in clutter,” *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3207–3217, July 2007.

[4] S. P. Sira, Y. Li, A. Papandreou-Suppappola, D. Morrell, D. Cochran, and M. Rangaswamy, “Waveform-agile sensing for tracking,” *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 53–64, January 2009.

[5] S. P. Sira, A. Papandreou-Suppappola, and D. Morrell, *Advances in Waveform-Agile Sensing for Tracking*, A. Spanias, Ed. San Rafael, California: Morgan and Claypool Publishers, December 2008.

[6] J. J. Zhang, Q. Ding, S. Kay, A. Papandreou-Suppappola, and M. Rangaswamy, “Agile multi-modal tracking with dependent measurements,” in *Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, November 2010, pp. 1653–1657.

[7] S. Liu, S. Bhat, J. J. Zhang, Q. Ding, R. Narayanan, A. Papandreou-Suppappola, S. Kay, and M. Rangaswamy, “Design and performance of an integrated waveform-agile multi-modal track-before-detect sensing system,” in *Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, November 2011, pp. 1530–1534.

[8] D. J. Kershaw and R. J. Evans, “Optimal waveform selection for tracking systems,” *IEEE Transactions on Information Theory*, vol. 40, no. 5, pp. 1536–1550, September 1994.

[9] S. P. Sira, D. Morrell, and A. Papandreou-Suppappola, “Waveform design and scheduling for agile sensors for target tracking,” in *Asilomar Conference on Signals, Systems and Computers*, vol. 1, Monterey, California, November 2004.

[10] S. Suvorova and S. D. Howard, “Waveform libraries for radar tracking applications: Maneuvering targets,” in *40th Annual Conference on Information Sciences and Systems*, Princeton, New Jersey, March 2006, pp. 1424–1428.

[11] J. Candy, *Bayesian Signal Processing: Classical, Modern, and Particle Filtering Methods.* Hoboken, New Jersey: John Wiley & Sons, 2009.

[12] C. Capraro, I. Bradaric, G. T. Capraro, and T. K. Lue, “Using genetic algorithms for radar waveform selection,” in *IEEE Radar Conference*, Utica, New York, May 2008, pp. 1–6.

[13] A. W. Doerry, "Generating nonlinear FM chirp waveforms for radar," Sandia National Laboratories, Albuquerque, New Mexico, Tech. Rep. SAND2006-5856, September 2006.

[14] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. part i: Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, October 2003.

[15] H. L. V. Trees, *Detection, Estimation, and Modulation Theory: Radar-Sonar Signal Processing and Gaussian Signals in Noise.* Malabar, Florida: Krieger Publishing Company, July 1992.

[16] B. S. Bourgeois, "Using range and range rate for relative navigation," Tech. Rep., September 2007.

[17] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Systems*, vol. 29, no. 6, pp. 82–100, December 2009.

[18] T. E. Fortmann, Y. Bar-Shalom, M. Scheffe, and S. Gelfand, "Detection thresholds for tracking in clutter - a connection between estimation and signal processing," *IEEE Transactions on Automatic Control*, vol. AC-30, no. 3, pp. 221–229, March 1985.

[19] D. Musicki and M. R. Morelande, "Gate volume estimation for target tracking," in *Proceedings of the Seventh International Conference on Information Fusion*, Stockholm, Sweden, June 2004, pp. 28–32.

[20] H. Sun and M. Farooq, "Note on the generation of random points uniformly distributed in hyper-ellipsoids," in *Proceedings of the Fifth International Conference on Information Fusion*, Annapolis, Maryland, July 2002, pp. 489–496.

[21] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, March 2004.

[22] R. Niu, P. Willett, and Y. Bar-Shalom, "Matrix CRLB scaling due to measurements of uncertain origin," *IEEE Transactions on Signal Processing*, vol. 49, no. 7, pp. 1325–1335, July 2001.

[23] C. Rago, P. Willett, and Y. Bar-Shalom, "Detection-tracking performance with combined waveforms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 2, pp. 612–624, April 1998.

[24] L. Cohen, *Time-Frequency Analysis.* Upper Saddle River, New Jersey: Prentice Hall, 1995.

[25] X. Li, Z. Bai, and K. Kwak, "NBI suppression UWB system based on novel nonlinear chirp pulses," in *9th International Symposium on Communication and Information Technologies*, Incheon, Korea, September 2009, pp. 1167–1170.

[26] J. Stewart, *Calculus*, 5th ed., B. Pirtle, Ed. Belmont, California: Brooks/Cole-Thomson Learning, 2003.

[27] C. Houck, J. Joines, and M. Kay, “A genetic algorithm for functional optimization: A Matlab implementation,” Tech. Rep. 95-09, 1995.

[28] S. Das and P. N. Suganthan, “Differential evolution: A survey of the state-of-the-art,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, February 2011.

[29] J. Mor, B. Garbow, and K. Hillstrom, “Testing unconstrained optimization software,” *ACM Transactions on Mathematical Software*, vol. 7, no. 1, pp. 17–41, March 1981.

[30] R. Mendes and A. S. Mohais, “DynDE: a differential evolution for dynamic optimization problems,” in *IEEE Congress on Evolutionary Computation*, vol. 3, Edinburgh, Scotland, September 2005, pp. 2808–2815.

[31] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, “The unscented particle filter,” Tech. Rep., August 2000.

APPENDIX A

LIST OF ACRONYMS

2D two-dimensional

3D three-dimensional

4D four-dimensional

BEC bounding exponential chirp

BFP bootstrap particle filter

CP-FM cubic polynomial FM

CRLB Cramér-Rao lower bound

DE differential evolution

EO electro-optical

FIM Fisher information matrix

FM frequency-modulated

IF instantaneous frequency

MLE maximum likelihood estimator

MSE mean-squared error

PDA probabilistic data association

PF particle filter

PMSE predicted MSE

QPC quadratic parabolic chirp

QPPF quartic polynomial phase function

SDT scaled-difference transform

SNR signal-to-noise ratio

STAP space-time adaptive processing

TB time-bandwidth

TBD track-before-detect

UKF unscented Kalman filter

UPF unscented particle filter

UT unscented transform