Parameter selection in a fully adaptive tracking radar

Jonas Myhre Christiansen
Norwegian Defence Research Establishment
Kjeller, Norway
Email: Jonas-Myhre.Christiansen@ffi.no

Graeme E. Smith
The Ohio State University
Columbus, Ohio, USA
Email: smith.8347@osu.edu

***Abstract*—A fully adaptive radar framework has been proposed in recent publications, and this paper will implement the framework for a tracking radar. A set of cost functions are developed to balance the performance cost of the track and the resource cost of the track update. The method and cost functions are illustrated in a simulated example which emulating a simple scenario. The performance of the radar is compared when adapting with the different cost functions, and the performance of the track is compared together with the resources spent. It showed an increased performance in track maintenance at less resources spent, when comparing the method adapting over track update interval, number of pulses in dwell and PRF, over a method only adapting over track update interval.**

I. Introduction

This paper reports on the use of the fully adaptive radar (FAR) framework [1] to implement an adaptive update-interval and parameter selection algorithm for a tracker. For the research conducted here, only single target tracking is utilized, however, the method could be applied to all tracks in a multiple target tracking case providing a path to true resource management. The method developed is validated using a MATLAB simulator. The results demonstrate that the FAR framework could effectively control update parameters for a tracker.

Advances in RF hardware, analog-to-digital converter (ADC), and computer processing technology have led to radars that exhibit significant diversity in their parameters. Where once the waveform duty-cycle, bandwidth and modulation were fixed, these parameters are now commonly thought of as configurable on a per dwell, if not per pulse, basis [2]. While it is certainly possible to use heuristics to make such decisions, the heuristic engine rapidly become unwieldy as the scale of the problem to be solved increases and sub-optimal solutions are produced [3]. Instead it is preferable to solve the problem of parameter selection using optimization based techniques. Optimization of parameter selection for radar became its own area of research, commonly referred to as either fully adaptive radar or cognitive radar [4], [5].

For the research considered here, we are concerned with the FAR framework, proposed in [1] and experimentally demonstrated in [6] to provide parameter optimization. While its initial studies were completed using simulation, the FAR framework proposed by Bell & Smith was validated experimentally, first using prerecorded data [1] and then through implementation on a dedicated hardware platform [6]. Subsequently, there has been significant investigation into the framework demonstrating methods by which it can be extended as well as exploring its limitations [7]–[9]. This consolidated set of research has yielded a parameter optimization methodology that can be applied in support of target tracking activities.

Given the close similarity between radar resource management and FAR it is a natural progression to understand if FAR can be utilized in the resource management problem. The initial FAR paper [1] did consider a resource management problem. However, the study was only in simulation and the example was designed for demonstration rather than realism. With the advances that have been made to the FAR framework in subsequent research [8], it would seem reasonable to make the parameter to be adapted the track update interval, in a manner consistent with van Keuk, [10], and thus enable the start of meaningful resource management using the FAR framework. Other work adapting the track update interval could be found in the MESAR program [11]. Including parameters such as dwell time, through adjusting pulse repetition frequency (PRF) and number of pulse, is important in a resource management context since minimizing the dwell time will allow for more resources to be spent on other targets. The work presented in this paper will be a novel implementation of the FAR framework that balances resource usage against global tracking error. A comparison of cost functions weighing resources spent against a global track error will be shown.

The remainder of this paper is organized as follows. Section II presents the cost function from the cognitive radar (CR) framework. Section III describes the simulator developed and show some simulated results. Section IV shows simulation result, and concluding remarks are given in section V.

II. Fully adaptive radar framework for parameter selection in a tracking radar

A. Tracking model

The tracking model for a converted measurement linear Kalman filter (LKF) from [12] is implemented using the FAR framework. The motion model is given in Cartesian coordinates, and the measurement uncertainties is given in the measurement space of the radar which is the range and bearing space. The measurement covariance matrix in Cartesian coordinates is generated as proportional and rotated version

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/radar41533.2019.171303

of the matrix in measurement space. The state space used in the Kalman filter is represented by

$$\boldsymbol{x}_{k+1} = \boldsymbol{\phi}_k(T_k)\boldsymbol{x}_k + \boldsymbol{w}_k \tag{1}$$

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k \tag{2}$$

where $\boldsymbol{\phi}$ is the transition matrix and $\boldsymbol{H}$ is a matrix selecting the position variables x, y. The state variable $\boldsymbol{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T$. The noise variables is defined as $\boldsymbol{w}_k \sim \boldsymbol{N}(0, \boldsymbol{Q}_k(T_k))$, $\boldsymbol{v}_k \sim \boldsymbol{N}(0, \boldsymbol{R}_k(\Delta r_k, \Delta v_k))$ and T_k is the update interval. ϕ_k is the state transition matrix from the state at time k to $k+1$. It usually follows normal kinematics where distance moved from time k to $k+1$ is the velocity from the state space times the update interval T_k. For nearly constant velocity, velocity is regarded as the same for $k+1$ as for k plus noise. $\boldsymbol{Q}_k$ is the covariance matrix of the process noise, and hence describes the motion model for the tracker. The size of the diagonal elements of $\boldsymbol{Q}_k$ shows the probable range of motion model parameters such as acceleration and velocity, and the model used for $\boldsymbol{Q}_k$ can be found in [12].

$\boldsymbol{R}_k$ is the covariance matrix for the measurement model, and describes the measurement accuracy. The measurement $\boldsymbol{z}_k$ is the Cartesian position variables. The true measurement space is non-linear with regards to the Cartesian space, since the measurement is in range and bearing. The transition from Carthesian to range and bearing is as follows

$$r = \sqrt{x^2 + y^2} \tag{3}$$

$$\psi = \arctan\left(\frac{y}{x}\right) \tag{4}$$

Measurement uncertainty in the radar measurement space are obtained from the accuracy models for radar sensors [13, pp. 402-437], and the lower bound for measurement accuracies is

$$\sigma_R \geq \frac{\Delta R}{\sqrt{SNR}} \tag{5}$$

$$\sigma_\psi \geq \frac{\theta_3}{2\sqrt{SNR}} \tag{6}$$

where ΔR is the range resolution and θ_3 is the 3dB beamwidth of the antenna array.

The converted measurements [14] is given as the rotated measurement accuracies, where the cross range uncertainty is calculated as the arc length uncertainty.

$$R_{11} = r^2\sigma_\psi^2 \sin^2\psi + \sigma_r^2 \cos^2\psi \tag{7}$$

$$R_{22} = r^2\sigma_\psi^2 \cos^2\psi + \sigma_r^2 \sin^2\psi \tag{8}$$

$$R_{12} = \left(\sigma_r - \sigma_\psi^2\right) \sin\psi \cos\psi \tag{9}$$

The measurement covariance matrix $\boldsymbol{R}_k$ can therefore be defined as

$$\boldsymbol{R}_k = \begin{bmatrix} R_{11} & R_{12} \\ R_{12} & R_{22} \end{bmatrix}, r = r_k, \psi = \psi_k \tag{10}$$

B. Fully adaptive radar framework and tracker model

A general FAR framework for detection and tracking was introduced in [1] for the development of CR applications. It is based on the perception-action cycle defined in [15], where

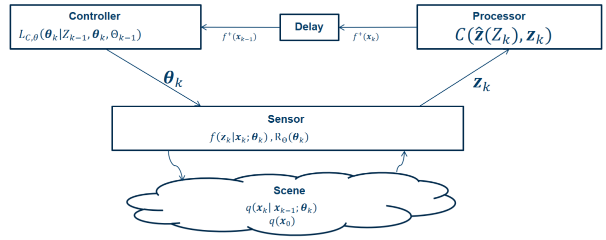

Figure 1. Diagram of cognitive radar framework

there is a feed-back loop between receiver and transmitter to allow for adaptation to the environment. The feed-back loop is illustrated in Figure 1, where the receiver is connected to the transmitter through a controller.

The Kalman filter recursion shown in [12] is used for motion and information update, where the predicted covariance matrix from the information update is equal to the predicted information matrix (PIM) [1] for a Gaussian density. The predicted conditional Cramér-Rao lower bound (PC-CRLB) is defined as the inverse of the predicted conditional Bayesian information matrix (PC-BIM), a concept defined in [1], where the PC-BIM is equal to the sum of PIM and expected value of the Fisher information matrix (FIM). The expected value of the FIM is defined as

$$\boldsymbol{J}_k^-(\boldsymbol{\theta}_k|\boldsymbol{Z}_{k-1}, \boldsymbol{\Theta}_{k-1}) = E_k\left\{\boldsymbol{J}_{\boldsymbol{x}}(\boldsymbol{x};\boldsymbol{\theta}_k)\right\} \tag{11}$$

where $\boldsymbol{\theta}_k$ is the optimizing parameters and $\boldsymbol{J}_{\bar{x}}$ is the FIM which is defined as

$$\boldsymbol{J}_{\boldsymbol{x}}(\boldsymbol{x};\boldsymbol{\theta}_k) = -E_{\boldsymbol{z}_k|\boldsymbol{x}_k;\boldsymbol{\theta}_k}\left\{\boldsymbol{\nabla}_x[\boldsymbol{\nabla}_x[\ln f(\boldsymbol{z}_k|\boldsymbol{x}_k;\boldsymbol{\theta}_k)]]^T\right\} \tag{12}$$

The probability density function (PDF) $f(\boldsymbol{z}_k|\boldsymbol{x}_k;\boldsymbol{\theta}_k)$ is a multivariate Gaussian distribution with covariance $\boldsymbol{R}_k$. The FIM is defined as the second partial derivative of the natural logarithm of the PDF, and can be shown to be

$$\boldsymbol{J}_{\boldsymbol{x}}(\boldsymbol{x};\boldsymbol{\theta}_k) = \boldsymbol{H}^T\boldsymbol{R}_k(\boldsymbol{\theta}_k)^{-1}\boldsymbol{H} \tag{13}$$

The expected FIM can be shown to be the time averaged expectation of the FIM.

$$\begin{aligned} J_k^-(\boldsymbol{\theta}_k|\boldsymbol{z}_{k-1}, \boldsymbol{\Theta}_{k-1}) &= E_k\left\{\boldsymbol{J}_{\boldsymbol{x}}(\boldsymbol{x};\boldsymbol{\theta}_k)\right\} \\ &= E_k\left\{\boldsymbol{H}^T\boldsymbol{R}_k(\boldsymbol{\theta}_k)^{-1}\boldsymbol{H}\right\} \end{aligned} \tag{14}$$

The PC-BIM is therefore given as

$$\boldsymbol{B}_k^{\uparrow}(\boldsymbol{\theta}_k|\boldsymbol{Z}_{k-1};\boldsymbol{\Theta}_{k-1}) = \boldsymbol{\Sigma}_k(\boldsymbol{\theta}_k)^{-1} + E_k\left\{\boldsymbol{H}^T\boldsymbol{R}_k(\boldsymbol{\theta}_k)^{-1}\boldsymbol{H}\right\} \tag{15}$$

where $\boldsymbol{\Sigma_k}$ is the predicted covariance matrix calculated from the Kalman filter. The PC-BIM has the property [1]

$$R_C^{\uparrow}(\boldsymbol{\theta}_k|\boldsymbol{Z}_{k-1};\boldsymbol{\Theta}_{k-1}) \geq tr\left\{\boldsymbol{B}_k^{\uparrow}(\boldsymbol{\theta}_k|\boldsymbol{Z}_{k-1};\boldsymbol{\Theta}_{k-1})^{-1}\right\} \tag{16}$$

For each iteration, the radar would then solve the minimization problem

$$\begin{aligned} \boldsymbol{\theta}_k &= \arg\min_{\boldsymbol{\theta}}\left[tr\left\{\left(\boldsymbol{\Sigma}_k(\boldsymbol{\theta})^{-1} + \boldsymbol{H}^T\boldsymbol{R}_k(\boldsymbol{\theta}_k)^{-1}\boldsymbol{H}\right)^{-1}\right\} + R_\Theta(\boldsymbol{\theta})\right] \\ & s.t. \boldsymbol{\theta} \in S \end{aligned} \tag{17}$$

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/radar41533.2019.171303

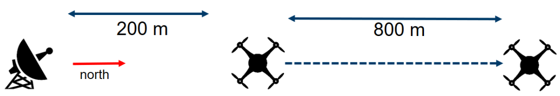

Figure 2. Simulation setup, where a target moves with constant velocity from the radar

where S is the set of possible parameters. This would usually be a set containing a continuous or discrete set of values within a lower and upper bound. The time averaging expectation operator E_k has been removed since the covariance matrix $\boldsymbol{R}_k(\boldsymbol{\theta}_k)$ is not a recursive function with respect to time step k, and the argument of the minimization algorithm is only dependent on current time step. $R_\Theta(\boldsymbol{\theta})$ is a function representing the resources spent on the track update. The part of (17) enclosed in the $tr...$ is the performance cost, and hence both resource and performance is optimized simultaneously. The following section will investigate candidate resource cost functions.

C. Candidate resource cost functions

Previous work by Christiansen and Smith [9] analyzed the performance of the resource cost equal to $\frac{A}{\theta}$, but with a tracker operating in the range and range rate dimension. A is a constant and θ is the track update interval. This article will analyze the performance of this cost function, when using a Cartesian tracker, and compare it to other candidate resource functions. The optimizing parameter for this function is track update interval only, and the cost function represents the resource usage of a track update as more costly for shorter track updates. It has an exponential growth, since the track updates spend the same amount of resource and the amount of available track updates greatly decreases when the track update interval is short. The resource cost function is given in (18), where θ_1 is the track update interval.

$$R_\Theta(\theta) = \frac{A}{\theta_1} \tag{18}$$

Next we consider a candidate resource function that accounts for the resources spent with the track update interval and includes the resources spent for the specific track update. A measure for resources spent could be the dwell time and it is represented as the number of pulses used in the dwell divided by the PRF. The second candidate function could therefore be a weighted sum of the resources spent with a specific track update interval and the dwell time. Equation (19) shows the resource cost function, where B is a constant, θ_2 is the number of pulses, θ_3 is the PRF and $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]^T$ is the parameter vector.

$$R_\Theta(\boldsymbol{\theta}) = \frac{A}{\theta_1} + B\frac{\theta_2}{\theta_3} \tag{19}$$

III. Fully adaptive radar simulator

A simulator has been implemented in Matlab to test applications using the FAR framework. The simulation parameters were selected to resemble the experimental radar system built at Norwegian Defense Research Establishment (FFI) [16] to test CR applications. The simulator has been used to develop and test CR algorithms prior to experiments on the experimental system. Both simulation parameters and a more detailed description of the simulator can be found in [9].

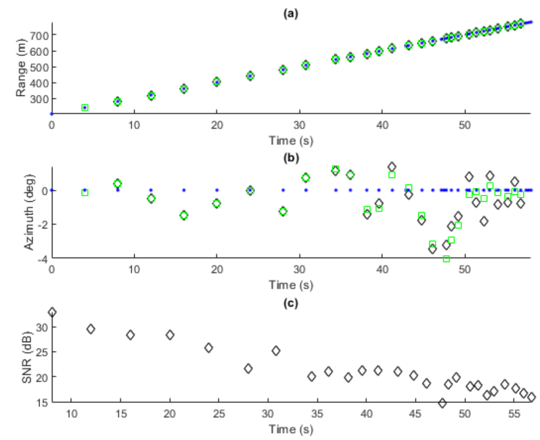

Figure 3. True position in blue dots, detections in black diamond and track in green square. (a) Radar range, (b) radar azimuth, (c) SNR

IV. Simulation results

This section will present results of simulations, using the tracker defined in section II-A and solving the optimization problem defined in section II-B, (17). Two candidate resource cost functions will be compared, given in (18) and (19). The simulated target has a simple movement with linearly increasing range and constant velocity which is shown in Figure 2. It moves northward in respect to the radar, as shown in Figure 2, and hence the azimuth angle is zero and it is moving along the y axis in local Cartesian coordinates. This is shown in Figures 6 (a) and 3 (a) which shows the target range for the simulations. Figures 6 (b) and 3 (b) shows the target azimuth, which is close to zero for the full simulations, as expected for a target moving northwards, and Figures 6 (c) and 3 (c) shows the signal to noise ratio (SNR) which decreases as the targets moves with increasing range.

Simulation parameter	Static value case	Bounds
Peak power	10 Watts	$<$10 Watts
PRF	10kHz	$<$ 100 kHz
CPI	51ms	$<$ 1 second
Track update rate		$<$ 8 Hz
Bandwidth	50MHz	160MHz
Angular accuracy		$>$ 4° (SNR dependent)
Beam width		30°
Center frequency	3.1GHz	3.0 - 3.3GHz

Table I
Key parameters for simulator of cognitive radar system

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/radar41533.2019.171303

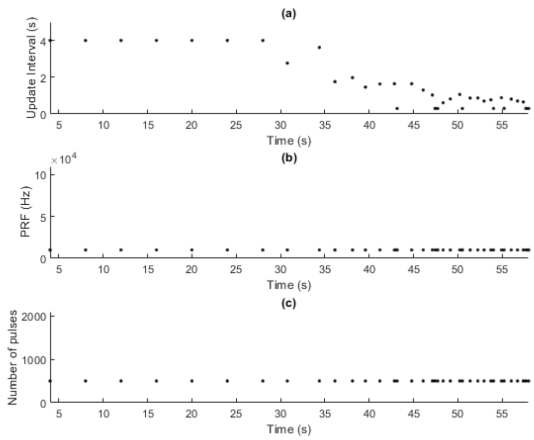

Figure 4. Radar parameters; (a) track update interval, (b) PRF, (c) number of pulses for dwell

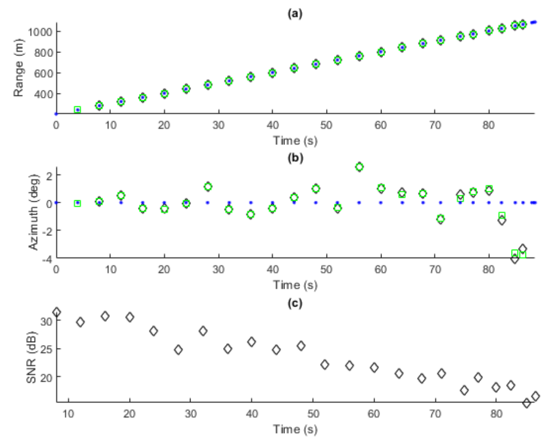

Figure 6. True position in blue dots, detections in black diamond and track in green square. (a) Radar range, (b) radar azimuth, (c) SNR

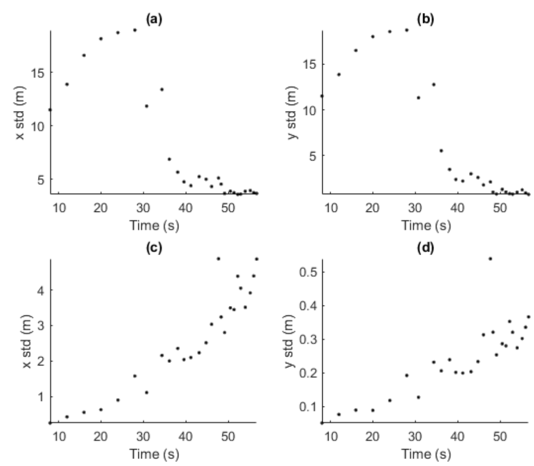

Figure 5. Tracker standard deviation; (a) prior x std, (b) prior y std, (c) measurement R_k for x std, (d) measurement R_k for y std

A. *Track update interval-only resource cost function analysis*

When using the resource cost function given in (18) to solve the optimization problem given in (17), the optimizer will balance between the predicted global track error from the trace of the PC-BIM and the track update interval. The predicted posterior covariance from the motion model of the tracker will be large for large track update intervals, since the uncertainty of where the target is will increase when the revisit time increases. The measurement uncertainty is dependent on the resolution in range and angle, combined with the SNR as shown in equations (5) and (6), and hence is not dependent on the optimizing parameter which is the track update interval.

Figure 4 shows how the radar parameters change over the simulation. (a), which show the track update interval, is the only relevant parameter since it is the only changing parameter from the resource cost function. When the target is close to the radar, the update interval is equal to the maximum allowed value of 4s. When the target moves away from the radar it decreases, because the cost function balances the state uncertainty dependent on the track update interval against the measurement uncertainty which is dependent on the SNR and range. The track is eventually lost after the track update has been the minimum allowed for a period of time, and this is due to low SNR which cause lost detections.

Figure 5 shows the track covariance in sub-figures (a) and (b) for x and y coordinates. The covariance is constant, after a short period of stabilizing, and this is due to the constant track update interval in the beginning. When the track update interval decreases, the covariance decreases as well, and this is mainly due to that the uncertainty of the target position due to the movement model from the tracker being dependent on the track update interval solely. The global covariance from the PC-BIM is dependent on both the track covariance and the measurement covariance, and hence it balances the two, against the cost of the track update. This means that when the measurement uncertainty increases, shown in (c) and (d) in figure 5, it has to compensate for this with decreasing the tracker error. This can be achieved with lowering the track update interval. One obvious observation is that the covariance in (c) is much larger than (d), and this is due to that the target moves straight along the antenna bore-sight, along y-axis, and the uncertainty is much larger in the angle measurement than in the range measurement for this system. (c) shows x uncertainty, and hence it shows the cross-range uncertainty which is approximately proportional to the angular uncertainty.

B. *Track update interval and dwell time resource cost function analysis*

When incorporating the dwell time into the optimization, two extra radar parameters is adjusted. The dwell time is the number of pulses divided by the PRF, and hence both parameters are included in the analysis. Figure 7 shows the radar parameters, and now we can see that all three parameters are adjusted during the simulation. The track now lasts for

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/radar41533.2019.171303

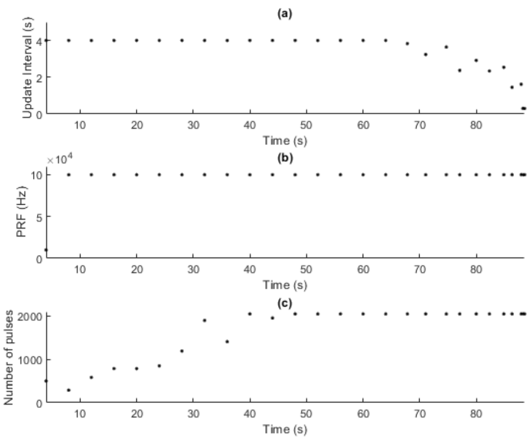

Figure 7. Radar parameters; (a) track update interval, (b) PRF, (c) number of pulses for dwell

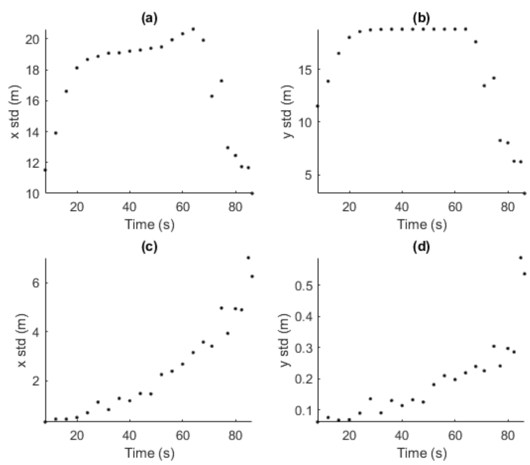

Figure 8. Tracker standard deviation; (a) prior x std, (b) prior y std, (c) measurement R_k for x std, (d) measurement R_k for y std

longer at the maximum allowed value of 4s (Figure 7, (a)), the PRF jumps to the maximum allowed value of 100kHz (Figure 7, (b)) and the number of pulses per Coherent Processing Interval (CPI) increases from 500 to 2000 (Figure 7, (c)) during the first part of the simulation. Increasing the number of pulses increases the SNR, and increasing the PRF decreases the dwell time with the same number of pulses. Figure 8 (a) and (b) shows the track covariance, and it is connected to the track update interval as in the example in the previous section. The measurement covariance, shown in Figure 8 (c) and (d), decreases more slowly than in the previous example, and this is due to the increased number of pulses. Even though more pulses are utilized, the dwell time is actually smaller than in the previous example since the starting PRF is 10kHz, a tenth of the adjusted PRF in this example. We could of course select such a PRF for the previous example, but the purpose here is to show that having adjustable parameters selected from an optimization makes the system more adaptable. In this example, the track is maintained longer at the expense of less resources, if we look at track update interval and dwell time. The second example utilizes more pulses per CPI, but it does not necessary require a higher resource cost since the CPI per dwell is smaller due to the high PRF.

The track is maintained until approximately 1000 meters (Figur 6 (a)), where the track in the previous example is maintained until approximately 700 meters (Figure 3 (a)). This is more than 40% increase in range. From Figure 7 (a), the track update interval starts decrease from it's maximum allowed value of 4 seconds after almost 70 seconds, and the track update interval in the previous example starts decreasing after 30 seconds (Figure 4 (a)). This is more than a doubling of the time, and hence range since it has a constant velocity, where it can operate at a large update interval. This has the potential to save time and resources for a tracking radar.

V. Concluding remarks

This work has shown the an application for the FAR framework of parameter optimization in a Cartesian tracking radar. There is a comparison of two cost functions, where one only optimizes the resources used by track update interval, and the second method optimizes track update interval and dwell time represented by the number of pulses and the PRF. A performance increase has been showed for the cost function optimizing over both track update interval and dwell time, and it shows that we can sustain a track longer with less resources spent than the case of only track update interval. The track is maintained for 40% longer range and the tracker can operate at the maximum allowed track update interval at more than double range compared to the first cost function. This has the potential to save time and resources for a tracking radar.

References

[1] K. L. Bell, C. J. Baker, G. E. Smith, J. T. Johnson, and M. Rangaswamy, "Cognitive Radar Framework for Target Detection and Tracking," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4553, no. c, pp. 1–1, 2015.

[2] D. Garmatyuk, J. Schuerger, and K. Kauffman, "Multifunctional software-defined radar sensor and data communication system," *IEEE Sensors Journal*, vol. 11, no. 1, pp. 99–106, Jan 2011.

[3] A. B. Charlish, "Autonomous agents for multi-function radar resource management," Ph.D. dissertation, UCL, 2011.

[4] J. R. Guerci, "Cognitive radar: A knowledge-aided fully adaptive approach," in *2010 IEEE Radar Conference*, May 2010, pp. 1365–1370.

[5] S. Haykin, "Cognitive radar: a way of the future," *IEEE Signal Processing Magazine*, vol. 23, no. January, pp. 30–40, 2006.

[6] G. E. Smith, Z. Cammenga, A. Mitchell, K. L. Bell, J. Johnson, M. Rangaswamy, and C. J. Baker, "Experiments with cognitive radar," *IEEE Aerospace and Electronic Systems Magazine*, vol. 31, no. 12, pp. 34–46, 2016.

[7] L. Ubeda-Medina and J. Grajal, "Implementation of the fully adaptive radar framework: Practical limitations," in *2017 IEEE Radar Conference, RadarConf 2017*, 2017.

[8] A. E. Mitchell, G. Smith, K. L. Bell, A. Duly, and M. Rangaswamy, "Hierarchical Fully Adaptive Radar," *IET Radar, Sonar & Navigation*, vol. 12, no. February, p. 12, 2018. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/iet-rsn.2018.5339

[9] J. M. Christiansen, K. E. Olsen, and G. E. Smith, "Fully adaptive radar for track update-interval control," in *2018 IEEE Radar Conference (RadarConf18)*, April 2018, pp. 0400–0404.

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/radar41533.2019.171303

[10] G. Van Keuk and S. S. Blackman, “On phased-array radar tracking and parameter control,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 1, pp. 186–194, 1993.

[11] S. Noyes, “Calculation of next time for track update in the MESAR phased array radar,” in *IEE Colloquium on Target Tracking and Data Fusion*, vol. 1998. IEE, 1998, pp. 2–2. [Online]. Available: http://digital-library.theiet.org/content/conferences/10.1049/ic_19980420

[12] S. S. Blackman, *Multiple-target tracking with radar applications*. Artech House, 1986.

[13] D. K. Barton, *Modern radar system analysis*. Artech House, 1988.

[14] D. Lerro and Y. Bar-Shalom, “Tracking With Debiased Consistent Converted Measurements Versus EKF,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 3, pp. 1015–1022, 1993.

[15] J. M. Fuster, “The cognit: A network model of cortical representation,” *International Journal of Psychophysiology*, vol. 60, no. 2, pp. 125–132, 2006.

[16] J. M. Christiansen, G. E. Smith, and K. E. Olsen, “USRP based cognitive radar testbed,” in *2017 IEEE Radar Conference, RadarConf 2017*, 2017.

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/radar41533.2019.171303