Chapter 7

Sense Smart, Not Hard: A Layered Cognitive Radar Architecture

Stefan Brüggenwirth, Marcel Warnke,
Christian Bräu, Simon Wagner, Tobias Müller,
Pascal Marquardt and Fernando Rial

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.71365

Abstract

In this chapter, we present a cognitive radar architecture based on the three-layer model by Rasmussen. The skill-based-layer is characterized by adaptive signal-processing approaches and target matched waveforms. The rule-based-layer comprises reactive execution of optimal illumination policies and resource-management. The knowledge-based layer allows for long term, goal-oriented mission- and trajectory planning. Each layer is illustrated by example algorithms and applications for implementation.

Keywords: adaptive filters, cognitive systems, closed-loop controllers, robotics, signal processing, system architectures

1. Introduction

Modern multifunctional radars with electronic beam-steering (AESA) provide many degrees of freedom to point the antenna beam, usage of the electromagnetic spectrum and waveform selection (**Figure 1**). Complex surveillance and reconnaissance scenarios require increased automation and suited man-machine-interfaces, which is enabled by the cognitive radar approach [1–3].

In this article we explain a cognitive radar architecture developed at the Fraunhofer FHR based on the three-layer-model of Rasmussen [4]. In the following, we will first introduce the concept of cognitive automation and derive our cognitive radar architecture. For each cognitive subfunction several technologies for realization are discussed and illustrated by example applications.

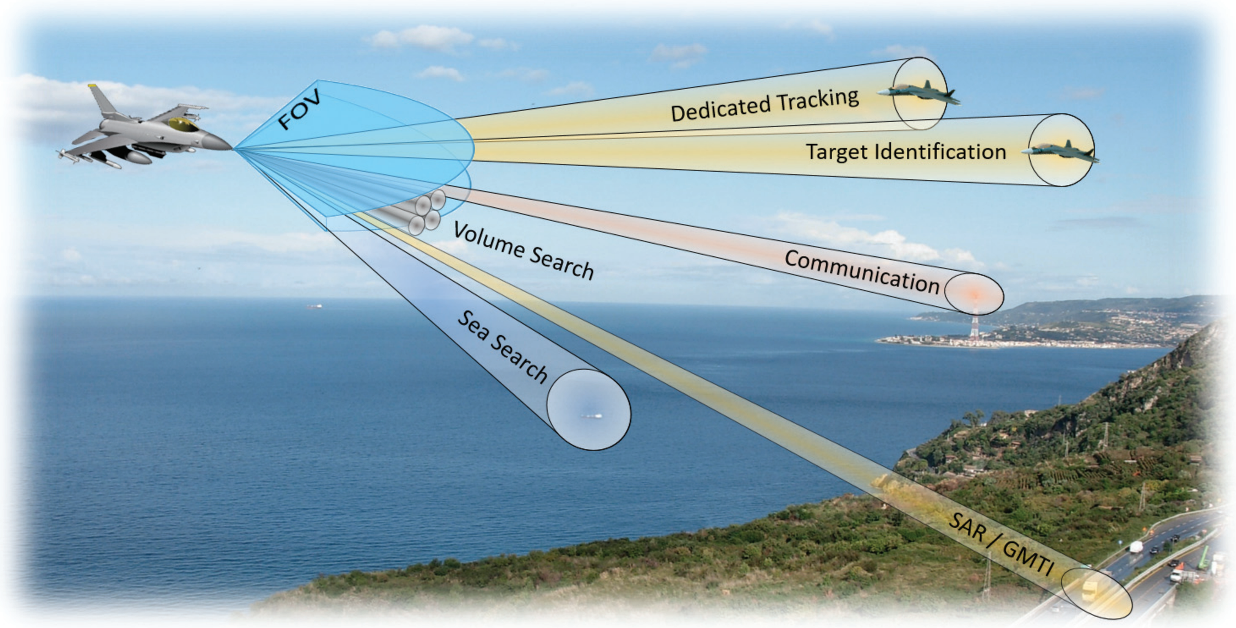


Figure 1. Airborne multifunctional radar.

2. Cognitive automation for radar

The concept of Dual-Mode Cognitive Automation [5] is well suited to deal with the challenges of highly automated radar systems. As shown in **Figure 2**, intelligent software-agents (depicted as robot-heads) can be introduced into the work equipment to increase the level of automation under the supervisory control paradigm [6] .

Alternatively the software-agent can cooperate with the human operator in the sense of an intelligent assistant system [7]. Even though the cognitive radar architecture can be used for both approaches, we will focus on the more traditional supervisory control role in the following.

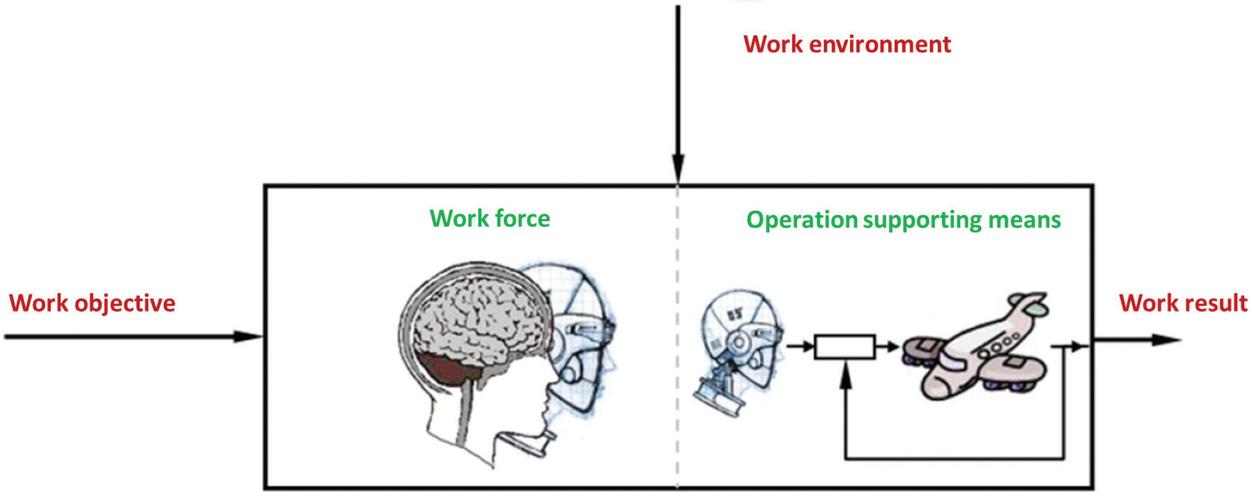


Figure 2. Concept of dual-mode cognitive automation [5].

3. Three layer model of a cognitive radar architecture

The three-layer model of human cognitive performance published by Jens Rasmussen in 1983 is widely used in human factors [8], cognitive psychology and robotics [9, 10]. As shown in **Figure 3** the complex process of human cognition is simplified and broken down into cognitive subfunctions (shown as gray boxes) with the indicated flow of information. The Rasmussen-model distinguishes three layers of cognitive performance with increasing level of abstraction.

The *skill-based-layer* comprises subconscious and very efficient perception and control tasks (such as steering along a curvy road). Above it, the *rule-based-layer* describes reactive behavior. Learned procedures are triggered by certain cues in familiar situations (such as stopping the car at a red traffic light). The *knowledge-based-layer* enables deliberate, goal-based behavior. By inferring novel solutions from a-priori knowledge flexible reaction in unknown situations is achieved (e.g. bypassing a traffic jam based on a road-map).

For the development of a cognitive radar architecture in analogy to the Rasmussen-model, each cognitive subfunction had to be mapped into five different radar-technologies as shown in **Figure 3**.

Modern radar system can generate arbitrary waveforms in real-time. This allows for transmit signals to be matched to the target transfer function or the electromagnetic spectrum as explained in Sections 4.1 and 4.2. Perception tasks of a radar comprise signal-processing and classification aspects. We use a machine learning approach that is illustrated in Section 4.3. Rule-based behavior in a radar is emulated by using optimal control policies or resource management approaches as shown in Section 5.1 or Section 5.2. Knowledge-based behavior can be implemented using Bayesian networks or automated planning algorithms. We show an example for robot-trajectory planning in Section 6.1.

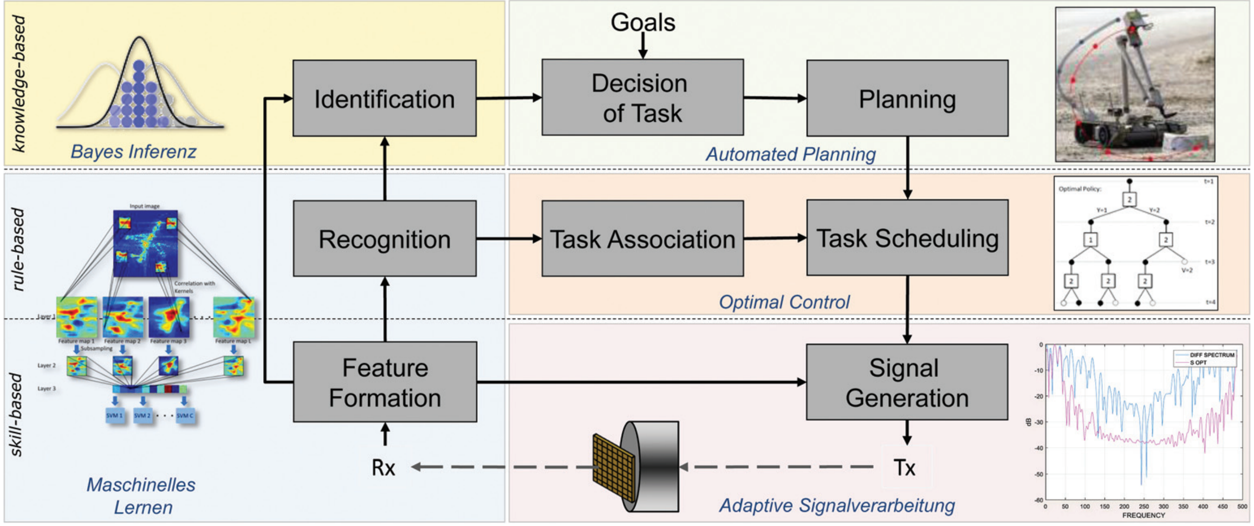


Figure 3. Three-layer-model of a cognitive radar architecture with supporting technologies. Modified from Ref. [4].

4. Skill-based-layer

The skill-based layer represents the basic signal-generation and processing capabilities of the radar system. It operates on the smallest timescale in the architecture in a continuous processing loop. Below, we give an example for adapting the transmit waveform to the target-transfer function using arbitrary waveform generation capabilities. As an extension, the waveform can further be adapted to the electromagnetic spectrum that has to be continuously sensed.

4.1. Matched illumination

If a priori information about a target is available, it is possible to optimize the transmission waveform for this target. Advantages arise for example by discriminating two classes of targets or by reducing resources of the sensor. One example is the reduction of the required bandwidth, if the available a priori information about the target is comprehensive.

In order to resolve the size of the object, two transmission frequencies are sufficient to estimate the extension of two scattering points with a spacing of Δz [11] (see **Figure 4**). The maximal energy at the receiver can be achieved when two frequencies are superposed to a beat frequency where the envelope covers the dimension of the target. If there are more than two scattering points the frequency spacing must be higher to achieve a higher period of the beat. In practice, the assumption of a known target impulse response is often difficult to realize. In a cognitive radar system, the a priori knowledge of the target can be presupposed by previous measurements and is assumed to be predicted for the next time step. An adapted waveform can be used to update the target track with respect to its extension by a lower allocation of the bandwidth.

Assuming a linear, time invariant channel with additive white Gaussian noise w, the complex received signal y corresponds to a convolution of the transmission signal s and the target transfer function h_i

$$y = y_{s,i} + w = h_i{}^{*}s + w = H_i s + w \tag{1}$$

where $y_{s,i}$ represents the undisturbed signal component. The linear convolution can be expressed by a matrix vector multiplication where the Toeplitz structured convolution matrix

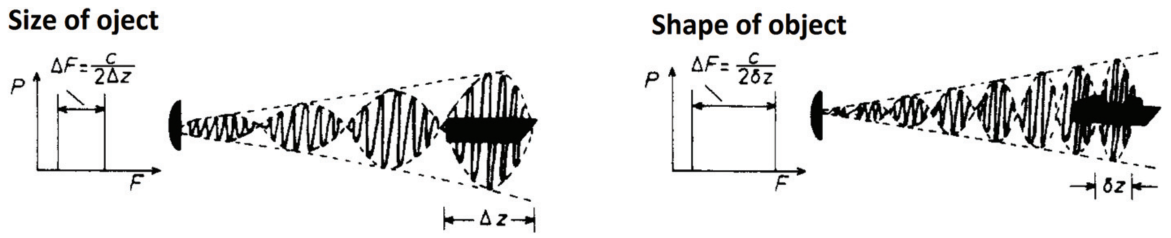

Size of oject
$\Delta F = \frac{c}{2\Delta z}$
P
F
Δz
Shape of object
$\Delta F = \frac{c}{2\delta z}$
P
F
δz

Figure 4. Transmitting two frequencies with a spacing of $\Delta F = c/2\Delta z$, the size of the object can be obtained. The shape of the object requires an even larger frequency separation $\Delta F = c/2\delta z$ (modified from ref. [11]).

H_i for the target index i is created by elements of the impulse response h_i. The detection performance is directly related to the signal to noise ratio (SNR) and depends on the receiver bandwidth and the power of the received signal y_s

$$\mathrm{SNR} = \frac{\|y_s\|_2^2}{\mathbb{E}\left[|w|^2\right]} = \frac{s^H H_i^H H_i s}{\sigma_w^2}. \tag{2}$$

If the target characteristic H_i is known, the signal to noise ratio can be increased by optimizing the waveform s [12]. The optimisation problem can be formulated to

$$\max_s \mathrm{SNR} = s^H A s \qquad \text{subject to} \qquad E_s = s^H s = \|s\|_2^2 = 1 \tag{3}$$

with the constraint of an energy limited transmission signal and the Hermitian correlation matrix $A = \frac{1}{\sigma_w^2} H_i^H H_i$. One possibility to solve this optimisation problem is the Lagrangian multiplier method

$$\begin{aligned} L(s,\lambda) &= s^H A s - \lambda\left(s^H s - 1\right) \\ \frac{\partial}{\partial s} L(s,\lambda) &= 0 = s^H A - \lambda s^H \Rightarrow (A - \lambda I)s = 0 \\ \frac{\partial}{\partial s^*} L(s,\lambda) &= 0 = s^T A^T - \lambda^T s^T \Rightarrow (A - \lambda I)s = 0. \end{aligned} \tag{4}$$

Eq. (4) is obviously an eigenvalue equation where the Lagrange multiplier λ represents the real eigenvalue and the waveform s is the corresponding eigenvector. By choosing the maximal eigenvalue, the signal to noise ratio is maximized. The eigenvector which corresponds to the maximal eigenvalue is directed towards the highest energy (variance).

To gain a better understanding of the solution, the basic example of **Figure 4** is presented for the two dominant scattering points

$$h(t) = a_0\delta(t - t_0) + a_1\delta(t - t_1) \qquad a_0 = |a_0|e^{j\varphi_0}, a_1 = |a_1|e^{j\varphi_1} \in \mathbb{C}. \tag{5}$$

The corresponding frequency spectrum to Eq. (5) is

$$\begin{aligned} H(\omega) &= \frac{1}{2\pi}\left(|a_0|e^{j\omega t_0 + \varphi_0} + |a_1|e^{j\omega t_1 + \varphi_1}\right) \\ |H(\omega)|^2 &= |a_0|^2 + |a_1|^2 + |a_0||a_1|\cos\left(\omega(t_0 - t_1) + \left(\varphi_1 - \varphi_0\right)\right). \end{aligned} \tag{6}$$

The target impulse response fluctuates due to the interference of the scattering centers with a period of $t_1 - t_0 = \frac{2}{c_0}(r_1 - r_0)$ and depends therefore on the target dimension as already visualized in **Figure 4**. The phase shift of the second point target causes a shift of all frequency maxima (see also **Figure 5(a)**). Processing an eigenvalue decomposition according to Eq. (4) to obtain the optimal waveform for this example (see **Figure 5(b)**).

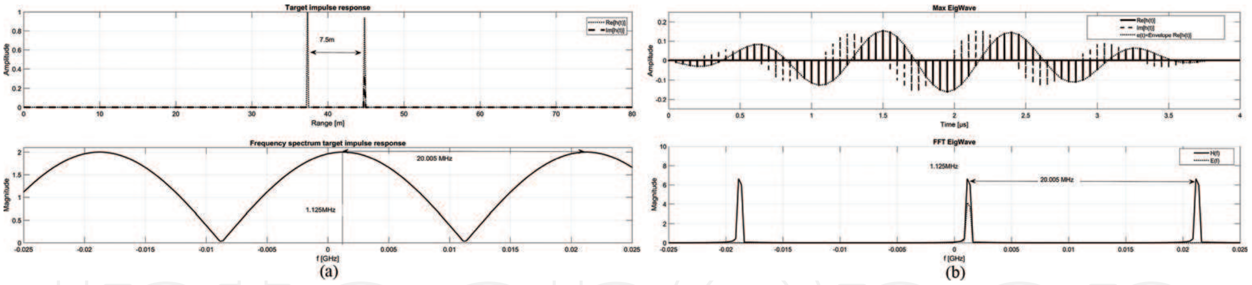


Figure 5. Target impulse response and optimal transmission waveform in time and frequency domain. (a) Target impulse response in time/range (upper) and frequency domain (lower) for two point targets at $r_0=37.32$ $(a_0=1)$ and $r_1=44.82$ $(a_1=1\angle 20^{\circ})$, (b) Optimal transmission waveform (eigenvector corresponding to the maximal eigenvalue) in time (upper) and frequency domain (lower).

Comparing this basic results with the solution of the eigenvalue decomposition, it is obvious that both frequency spectra are related to each other. If all frequency components of the target impulse response have comparable magnitudes, the frequency characteristic of the largest eigenvalue is similar to the target frequency spectrum and the target extension respectively. According to Eq. (6) the period of the target extension corresponds to a frequency of $\Delta F = \frac{c_0}{2(r_1 - r_0)} \approx 20.0$ MHZ. The envelope of the transmission wave is related to the phase difference between r_0 and $r_{1,}$ and causes a frequency shift in the frequency domain with $f_e = (\phi_1 - \phi_0)\frac{\varphi F}{2\pi} \approx 1.125$ MHZ. Summarizing the characteristic of the optimal transmission signal, the eigenvector corresponds to the main direction of the target variance in the frequency domain and is linked to the physical behavior of the target. If there are more than two scattering points, additional modulation products will occur. In the case where all frequency components of the target spectrum have similar magnitudes, the eigenvector corresponding to the largest eigenvalue will represent all constructive interferences in the resolution bandwidth. But also for small deviations of the spectral magnitudes, the main component (optimal eigenvector) will contain only the dominant frequency while the minor amplitudes are represented by the remaining eigenvectors forming finally the complete signal space.

In order to distinguish between targets an adapted waveform can be used to improve the discrimination between two types of target classes [2]. A binary hypothesis test is one method to discriminate between target classes by evaluating the received signal

$$\begin{aligned} \mathbb{H}_0 &: y = y_{s,0} + w = H_0 s + w \\ \mathbb{H}_1 &: y = y_{s,1} + w = H_1 s + w. \end{aligned} \tag{7}$$

The distance $d = \|y_{s,0} - y_{s,1}\|_2 = \|(H_0 - H_1)s\|_2$ denotes the difference of the received amplitude without taking noise into account. The robustness against incorrect classification increases for higher distances especially in a noise environment. Similar to Eq. (2)–(4), the optimal waveform can be calculated by solving

$$\begin{aligned} 0 &= (A_{01} - \lambda I)s_{opt} \\ A_{01} &= (H_0 - H_1)^H (H_0 - H_1). \end{aligned} \tag{8}$$

The energy is focused in the spectral area where the both target deviations are predominant.

Comparing the performance of a binary hypothesis test for a linear chirp and the optimized waveform, the test statistic of the likelihood ratio for Eq. (7) is calculated [13]

$$L = \log\left(\frac{p_{\mathbb{H}_0}}{p_{\mathbb{H}_1}}\right) = -\frac{1}{\sigma_w^2}\left(y - y_{s,0}\right)^H\left(y - y_{s,0}\right) + \frac{1}{\sigma_w^2}\left(y - y_{s,1}\right)^H\left(y - y_{s,1}\right) \tag{9}$$

with the variance σ_w^2 of the complex noise. The deflection for the likelihood ratio test defines the effective difference of the likelihood centers and represents the output signal to noise ratio

$$d = \frac{\left[\mathbb{E}_{\mathbb{H}_0}(L) - \mathbb{E}_{\mathbb{H}_1}(L)\right]^2}{Var_{\mathbb{H}_1}(L)} = \frac{1}{\sigma_w^2}\left(y_{s,0} - y_{s,1}\right)^H\left(y_{s,0} - y_{s,1}\right). \tag{10}$$

It is possible to achieve the same performance of the receiver operating characteristic (ROC) curve, with a different test statistic of that likelihood ratio but different deflections [14]. That is why a higher deflection is related to a better discrimination and a lower sensitivity with respect to an suboptimal threshold. **Figure 6** shows the results of the binary test for a linear frequency modulation (LFM) and the optimized waveform for two Gaussian targets with the same extension and distance. The deflection between both classes increases for the optimised waveform leading to a lower intersection are of the test statistic for the hypothesis and the alternative. This facilitates a better separability as well as a lower false alarm rate for the same detection probability.

One example of adapting the waveform to the environment is the support of the classification and saving resources like the bandwidth. Applications like interference mitigation can also be executed in the skill-based layer by combining spectrum sensing algorithms with matched illumination.

4.2. Spectrum sensing

Due to the fact that wireless communication technologies are of significant importance in modern times, the available radio frequency spectrum has become a valuable resource for radar. For example the U.S. department of commerce [15] has decided to allocate parts of the S-band (1695–1710 MHZ and 3550–3650 MHZ) to wireless communication. Another example are parts of the C-band (5150–5350 MHZ and 5470–5725 MHZ) which are used by weather

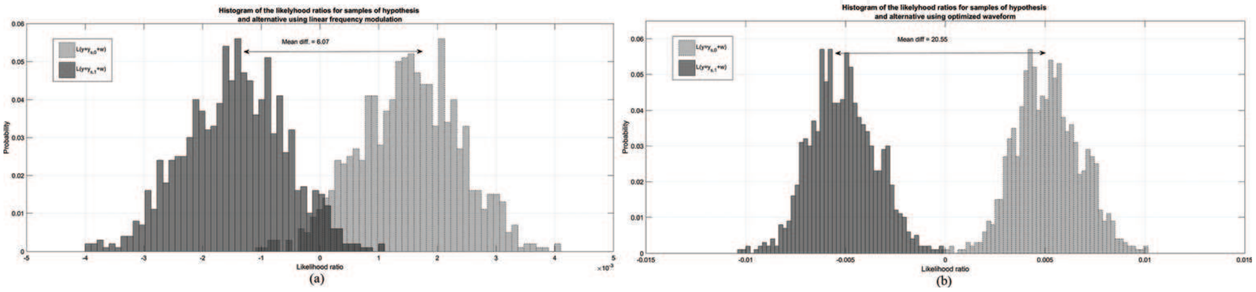


Figure 6. Test statistic of the likelihood ratios with the mean distance of the centers for LFM and optimized waveform. (a) Distribution of the likelihood ratio for noisy samples of the hypothesis and alternative using linear frequency modulation. (b) Distribution of the likelihood ratio for noisy samples of the hypothesis and alternative using the optimized waveform.

radars but are also used by 5GHz-WiFi [16] now. On the other hand these bands, although allocated, are underutilized providing opportunities for secondary (unlicensed) users to share the bands without harming the primary users. The other way round, a similar problem arises when the radar suffers interference from other users or even active jamming. Especially the first is a problem for ultra wideband radars like ground penetrating radars which naturally operate in partially occupied frequency bands. In the future these problems will become even worse and hence future cognitive radar systems must be able to operate in spectrally dense environments. Spectrum sensing techniques from cognitive radio provide algorithms to identify spectrum opportunities, i.e. to decide if a frequency band is occupied or not. With this information a cognitive radar can adapt dynamically its bandwidth, frequency and other transmit parameters to the radio frequency environment.

A significant number of studies dealing with spectrum sensing algorithms exists and hence we only give a brief overview here. For a comprehensive overview the reader is referred, for example, to the surveys [17, 18]. Spectrum sensing algorithms can be split into wideband and narrowband algorithms. Almost all narrowband spectrum sensing methods are statistical hypothesis tests usually written as

$$\mathbb{H}_0 : x(t) = w(t) \quad \text{against} \quad \mathbb{H}_1 : x(t) = s(t) + w(t)$$

where $x(t)$ represents the received complex signal, $s(t)$ the signal of another user and $w(t)$ the noise which is usually assumed white and Gaussian with variance σ_w^2. The most simple spectrum sensing method is the energy detector

$$\frac{2}{\sigma_w^2}\sum_{k=1}^{n} |x_k|^2 \underset{\mathbb{H}_0}{\overset{\mathbb{H}_1}{\gtrless}} F^{-1}_{\chi^2_{2n}}\left(1 - p_{fa}\right) \tag{11}$$

where p_{fa} is the desired probability of false alarm and $F_{\chi^2_{2n}}$ is the χ^2 distribution function with $2n$ degrees of freedom. Although this method is easy and fast, it suffers from bad detection probabilities in low SNR regions and poor robustness, see [19]. More advanced methods exploit certain features like for example cyclostationary properties [20] where a time series x_1, x_2… is said to exhibit cyclic frequency α with delay m if

$$C(\alpha, m) := \lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n} \text{red}\ \mathbb{E}[x_k x_{k+m}]e^{-j\,2\pi\alpha k} \neq 0. \tag{12}$$

Most modern modulations like OFDM or QAM have cyclostationary properties. For details on a test statistic see [21]. These methods offer high detection probabilities even in low SNR regions and are blind in the sense that they do not need information about σ_w^2. The price is a very high computational complexity and prior information about the used modulation. Completely blind methods, i.e. absolutely no prior information is necessary, are based usually on a multi antenna system. The data from the different channels is used to estimate a covariance matrix and from its characteristics e.g. eigenvalues a test statistic is build, see [22].

Because the channel state may change between the sensing and transmitting a prediction step after the sensing is helpful or even needed. For this purpose hidden Markov models are used in Ref. [23] and additionally multilayer perceptrons and recurrent neural networks are considered in Ref. [24]. Especially the neural networks perform well in simulations with a prediction accuracy of about 0.8 to 0.9.

In contrary to the narrowband band spectrum sensing the wideband spectrum sensing methods divide a band into occupied and unoccupied subbands. The most obvious method for classifying a wideband is to split it into fixed subbands (using a FFT or sweep and tune) and perform narrowband sensing in each one. But there are also native wideband spectrum sensing methods like a wavelet based approach, see [25].

If the radar is the primary user and avoiding or reducing interference is the only goal of the spectrum sensing, it is not necessary to decide if a channel is occupied or not. It is sufficient to use the channel with the least interference. But if a lot of interference is present, a compromise between bandwidth (resolution) and interference must be made which leads to an optimisation problem, see Refs. [26, 27].

After each sensing period, a suitable and adaptable waveform must be generated taking the information from the sensing step into account, essentially bandwidth and center frequency. For example, this can be multiple or notched chirps filling the unoccupied bands or a stepped FM waveform which avoids the occupied frequencies, see [27]. A combination with the matched illumination approach presented in Section 4.1 can be considered, too.

Building an experimental radar system with spectrum sensing capabilities is a challenging task. The computational complexity of some algorithms can be a burden and the additional sensing time, i.e. gathering the samples and computation time must be taken into account, causing a reduced duty cycle or pulse repetition frequency. In Ref. [27] a radar system employing spectrum sensing and matched illumination was implemented using an Ettus USRP X310 software defined radio. In a test environment about 10 dB noise floor reduction were achieved using spectrum sensing and a notched chirp.

4.3. Classification with deep learning techniques

The transition from the continuous stream of incoming row-data towards a symbolic representation of objects, which forms the basis for higher-level cognitive processing, is typically achieved using pattern recognition or classification techniques. As shown in **Figure 3**, machine learning approaches comprise subsymbolic feature formation processes that separate characteristic signal features in a higher-dimensional space. In this feature space, it is easier to recognize certain target classes to create an abstracted situational picture within the cognitive radar system.

4.3.1. Convolutional neural networks

Convolutional Neural Networks (CNNs) are inspired by the visual system of the brain and are part of the deep learning research field. For many years, CNNs were the only type of deep neural network that could efficiently be trained due to their structure using the technique of

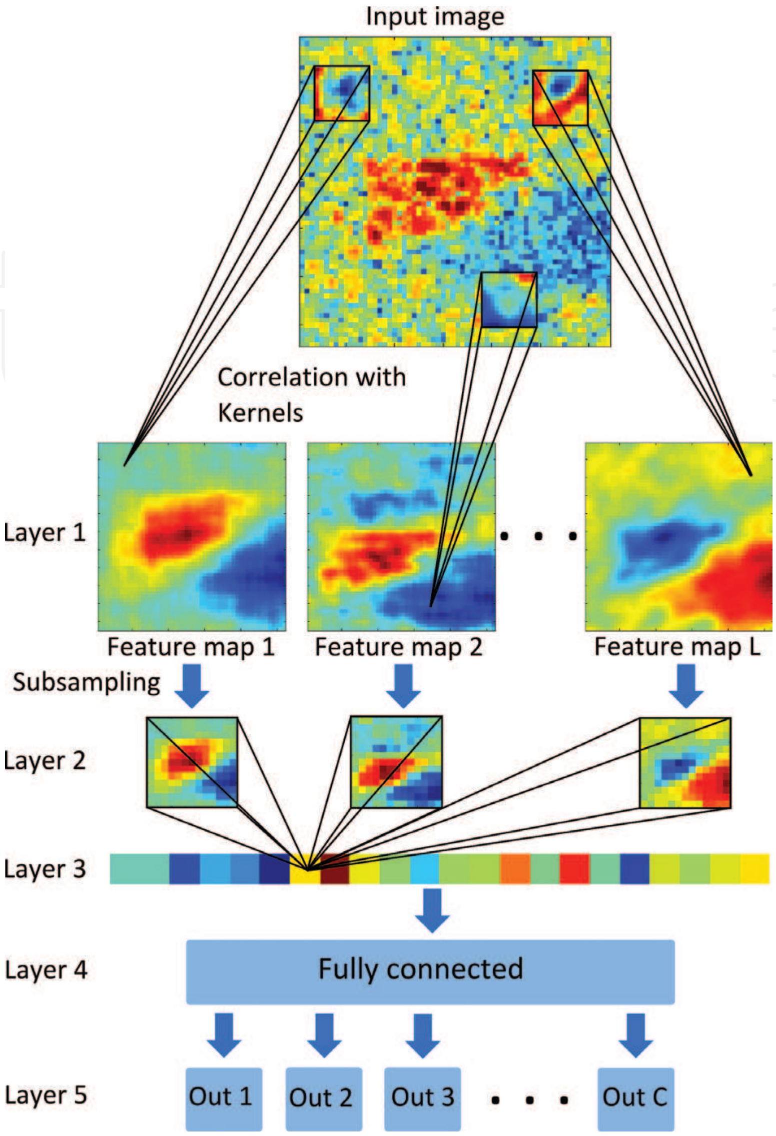


Figure 7. Structure of the used convolutional neural network.

weight sharing [28]. The basic structure of the network used in the presented architecture is shown in **Figure 7**.

CNN's are a special form of multi-layer perceptrons, which are designed specifically to recognize two-dimensional shapes with a high degree of invariance to translation, scaling, skewing, and other forms of distortion [29]. This invariance is achieved by an alternation of convolutional and subsampling layers, in which the neurons are organized in so called feature maps. All neurons in each of these feature maps use the same weights and are connected to a local receptive field in the previous layer. With this weight sharing technique, the number of free parameters is dramatically reduced compared to a fully connected network, what should lead to a better generalization of the network.

In the first convolutional layer, each neuron takes its inputs from a local receptive field in the input image and the output values of each feature map, which are visible in **Figure 7**, represent the intensity of one specific local spatial feature. The features, i.e. the weights of the neurons, are learned during the training process and since the receptive fields of neighboring neurons in the feature maps are shifted only by one pixel in the corresponding direction in the input image, the output values of each feature map correspond to the result of a two-dimensional correlation of the input image with the learned weights of each particular feature map.

In the input image of **Figure 7** one target is visible in the center of the image. The correlation with the different kernels is visualized for three examples. The learned kernels are depicted inside the black squares on the input image and the result of the correlation can be seen in the feature maps of the first layer.

The second layer of the network is a subsampling layer and performs a reduction of the dimension by a factor of four. With this reduction the exact position of the feature becomes less important and it reduces the sensitivity to other forms of distortion [29]. The subsampling is done by averaging an area of 4×4 pixels, multiplying it with a weight w_j and adding a trainable bias b_j.

The third layer is a convolutional layer again and relates the features found in the image to each other. This layer is trained to find pattern of features, which can be separated by the subsequent layers and discriminate the different classes. The output of this layer is the internal representation and can be considered as feature vector found by the network for the given input image.

The last two layers of the network form the decision part of the system and are fully connected layers, which use the output values of the third layer as features for classification. The last layer consists of as many neurons as classes have to be separated, in our case ten. The classification is done by assigning the corresponding class of the neuron with the highest output value.

One cost function for neural networks trained with the back propagation algorithm is the mean square error (MSE) of the training set. The MSE is the mean value of the quadratic loss function $E(\alpha)$, which is given by

$$E_i(\alpha) = (d_i - f(x_i, \alpha))^2. \tag{13}$$

In (13), α is the set of classifier parameters, d_i is the desired output for the ith element of the training set and $f(x_i, \alpha)$ is the classifier response to input x_i. The MSE of the complete training set with size N is thus

$$MSE(\alpha) = \frac{1}{N}\sum_{i=1}^{N} E_i(\alpha) = \frac{1}{N}\sum_{i=1}^{N} (d_i - f(x_i, \alpha))^2. \tag{14}$$

The MSE is also called the empirical risk with respect to quadratic loss and classifiers using this error as a performance measure are said to implement the empirical risk minimization (ERM) [30].

The training of our network is performed by the stochastic diagonal Levenberg-Marquardt algorithm that is presented in [31, 32]. The core of this algorithm is the stochastic update rule

$$\alpha_l^{(k+1)} = \alpha_l^{(k)} - \gamma_l^{(k)} \frac{\partial E_i(\alpha)}{\partial \alpha_l^{(k)}}, \tag{15}$$

where $\alpha_l^{(k)}$ is the l-th element of the parameter set α at iteration k, E_i is the instantaneous loss function of (13) for image i and $\gamma_l^{(k)}$ is the step size for the particular weight α_l at iteration k. The dependency of the step size on the iteration indicates that the step size is not fixed during the training, but is dynamically updated. The calculation of the step size is done by

$$\gamma_l^{(k)} = \frac{\eta^{(k)}}{\mu + g_l^{(k)}} \tag{16}$$

with the constant μ and a parameter $\eta^{(k)}$ that prevents the step size from becoming too large when the estimate of the second derivative $g_l^{(k)}$ of the loss function $E_i(\alpha)$ with respect to α_l is small. For the calculation of $g_l^{(k)}$ the Gauss-Newton approximation is used that guarantees a nonnegative estimate [32]. The parameter η is marked here as dependent on the iteration, but is fixed over several epochs of the training[1]. The Hessian matrix $g^{(k)}$ is not calculated explicitly in each iteration, instead a running estimate is kept that is updated with

$$g^{(k)} = (1-\beta)g^{(k-1)} + \beta \frac{\partial^2 E_i(\alpha)}{\partial \alpha^2}, \tag{17}$$

where β is between zero and one. Because of the weight sharing, the first and the second partial derivative of the loss function are sums of partial derivatives with respect to the connections that actually share the specific parameter α_l

$$\frac{\partial E_i(\alpha)}{\partial \alpha_l^{(k)}} = \sum_{(m,n)\in V_l} \frac{\partial E_i(\alpha)}{\partial w_{mn}^{(k)}}, \tag{18}$$

$$\frac{\partial^2 E_i(\alpha)}{\partial \alpha_l^{2(k)}} = \sum_{(m,n)\in V_l} \sum_{(j,k)\in V_l} \frac{\partial^2 E_i(\alpha)}{\partial w_{mn}^{(k)} \partial w_{jk}^{(k)}}. \tag{19}$$

In (18) and (19), the w_{mn} is the connection weights from neuron n to m and V_l is the set of unit index pairs (m, n) such that the connection between neuron m and n shares the parameter α_l, i.e.,

$$w_{mn} = \alpha_l \qquad \forall\ (m,n)\in V_l. \tag{20}$$

Further details of the algorithm and the approximations that are done to compute the derivatives can be found in Ref. [32].

[1] The training of neural networks is separated into epochs, in each epoch the complete dataset is presented one time to the classifier [29].

4.3.2. Regularizations and adaptive learning rates

One feature of the presented network is the use of *momentum*, which adds a feedback loop and with this some kind of memory to the algorithm. With this technique a certain amount of the weight change of the last iteration is added to the weight change of the current iteration. This amount is determined by the *momentum constant* ρ and leads to the expression

$$\Delta\alpha^{(k)} = \rho\Delta\alpha^{(k-1)} - \gamma^{(k)}\frac{\partial E_i^{(k)}(\alpha)}{\partial\alpha^{(k)}}, \tag{21}$$

which can also be written as

$$\Delta\alpha^{(k)} = \sum_{n=0}^{k-1}\rho^n\gamma^{(k-n)}\frac{\partial E_i^{(k-n)}(\alpha)}{\partial\alpha^{(k-n)}}. \tag{22}$$

The use of momentum should have a positive effect on the behavior of the training algorithm and may prevent the algorithm from converging to a local minimum of the error function [29]. Another important regularization method used in this network is the max-norm regularization of the weights of the network. For this regularization the Frobenius norm of each kernel in layer one and three is calculated after the weight change at every iteration and if the norm is larger than a certain value c, the kernel is rescaled to a norm of c. With this regularization an improvement of the convergence properties of the training algorithm has been observed.

So far the learning rate in (16) is only determined by the characteristics of the data itself and the error it produces at the output of the network. Another important factor could be meta-information available about the training set. We give here an example of a priority class, which means that we have one target in our database that should always be classified correctly with the additional cost that we might produce more errors in other classes. To incorporate these priority classes into our network, the representation in (22) is used. The general idea is to increase the learning rate $\gamma^{(k)}$ if an image of a priority class is presented at the current iteration. This is done by multiplying a priority weighting p with the learning rate $\gamma^{(k)}$, which is then marked as $\gamma^{'(k)}$

$$\gamma'^{(k)} = p\gamma^{(k)}. \tag{23}$$

If this term is included in the formula for the weight change $\Delta\alpha^{(k)}$, the sum in (22) can be split into two parts. One part that contains all samples of the priority classes and one part with the examples of the remaining classes

$$\Delta\alpha^{(k)} = \sum_{\substack{n\,\in\,priority\\ class}}^{k-1}\rho^n\gamma'^{(k-n)}\frac{\partial E_i^{(k-n)}(\alpha)}{\partial\alpha^{(k-n)}} + \sum_{\substack{n\notin priority\\ class}}^{k-1}\rho^n\gamma^{(k-n)}\frac{\partial E_i^{(k-n)}(\alpha)}{\partial\alpha^{(k-n)}}. \tag{24}$$

The need for a different weighting of classes is also discussed in Ref. [33], where it is mentioned that the different costs of misclassification should be part of the classifier design. The way we used here to include this prior knowledge into our target recognition system was also mentioned in Ref. [34] for Support Vector Machines, where the idea was to penalize the samples of less represented classes higher than others.

To show the benefit of this adaptive learning strategy we show an example of the ten class moving and stationary target acquisition and recognition (MSTAR) data [35] in **Figure 8**. In this example the learning rate of class four is multiplied with different weightings between one, which means no priority, and ten.

Without any weighting, this class has compared to the other classes a rather low correct classification rate calculated with respect to the number of input images $\mathrm{Pcc}_{\mathrm{in}}$ (curve with round markers). This value gives the amount of input images that belong to class four and are actually classified as class four. The curve with the square markers in the plot gives the probability of correct classification with respect to the number of output images $\mathrm{Pcc}_{\mathrm{out}}$, which gives the amount of images that are classified as class four really belong to class four and is thus an indicator on the reliability of the classification. Summarized over all classes, both indicators lead to the same result, the correct classification rate Pcc of the curve with the triangular markers. From the plot can be seen that $\mathrm{Pcc}_{\mathrm{in}}$ shows a steep increase at small values of p and up to $p=4$ also the overall correct classification rate increases, which is not the purpose here, but shows the positive effect of the additional correct classifications. While $\mathrm{Pcc}_{\mathrm{in}}$ is increasing, $\mathrm{Pcc}_{\mathrm{out}}$ shows a steady decreasing behavior. In the extreme case of $p \to \infty$, $\mathrm{Pcc}_{\mathrm{in}}$ should reach one and both $\mathrm{Pcc}_{\mathrm{out}}$ and Pcc should reach a value of N_{class4}/N, which means that all images in the dataset are classified as class four. This example and more details about the use of different weightings of different classes can be found in Ref. [36].

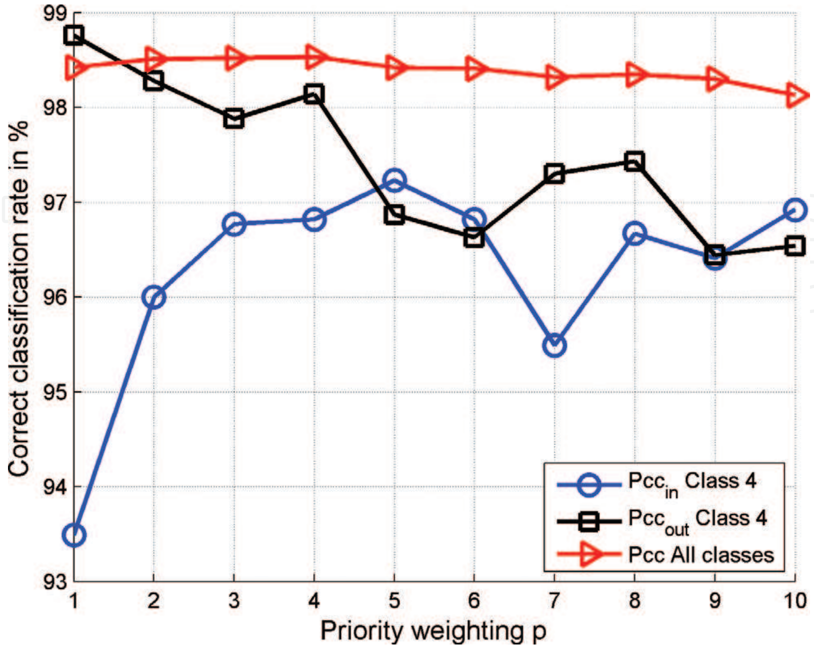


Figure 8. Performance of CNN with priority class.

4.3.3. Combination of convolutional neural networks with support vector machines

An often mentioned benefit of Support Vector Machines (SVMs) is the high generalization capability in comparison to neural networks. The high generalization of SVMs is achieved by a training strategy called *structural risk minimization*, which in comparison to the *empirical risk minimization* of neural networks takes the complexity of the classifier into account. For this reason, the Vapnik-Chervonenkis (VC)-dimension h was introduced to measure the complexity of a classifier. The VC-dimension is defined as the largest training set size N, which can be separated with binary labels in an arbitrary way by the SVM. With a high number of free parameters, the capacity of the classifier increases and thus the VC-dimension increases as well. Due to this relation, single patterns have a higher influence on the classification result for classifiers with a high VC-dimension, which increases the likelihood of overfitting to the training data [37]. To incorporate the VC-dimension into the minimization problem that has to be solved during the training, an additional term $\Phi\left(\frac{N}{h}\right)$ is added to the empirical risk to define the structural risk

$$R_{struc}(\alpha, h) = R_{emp}(\alpha, h) + \Phi\left(\frac{N}{h}\right), \tag{25}$$

where R_{emp} corresponds to the empirical risk. In this problem R_{emp} does not refer to the MSE of (14), which was used for neural networks, but to the specific number of misclassifications in the training set. The VC-dimension has an influence on both terms because a high VC-dimension will increase the complexity of the classifier and thus reduce the empirical risk, but the confidence interval $\Phi\left(\frac{N}{h}\right)$ would increase at the same time, since it only depends on the ratio between the size of the training set and the VC-dimension. SVMs are designed to find the best trade-off between these two terms, decrease the empirical error while keeping the VC-dimension as low as possible. Because of this, SVMs are classifiers with a very high generalization capability.

To use the high generalization of SVMs in our classification framework, we replace the last two layers of the CNN in **Figure 7** with SVMs. In this way we can use the convolutional feature extraction with the invariance to different forms of distortion and a classifier with high generalization. As input for the SVMs, the output values of the third layer are used. The final structure of the classifier is shown in **Figure 9**.

A SVM can only separate between two classes, for this reason the training set must be split for each SVM into two parts, one part containing the class that should give a positive result at the output of the SVM and one part containing the remaining training set that should give a negative result at the output. SVMs trained in that way are working in the one vs. all classification scheme, which means that as many SVMs as classes that need to be separated are necessary. For the actual classification of a SVM, a kernel is used to transform the data to a high dimensional space in which it is more likely that the problem can be linearly separated. Two common kernels are polynomial (including linear and quadratic kernels) and radial basis functions (RBFs). In **Table 1** a small example of the MSTAR database is shown and it can be seen that the already very high correct classification rate of the CNN can be further increased with the use of SVMs as classifier.

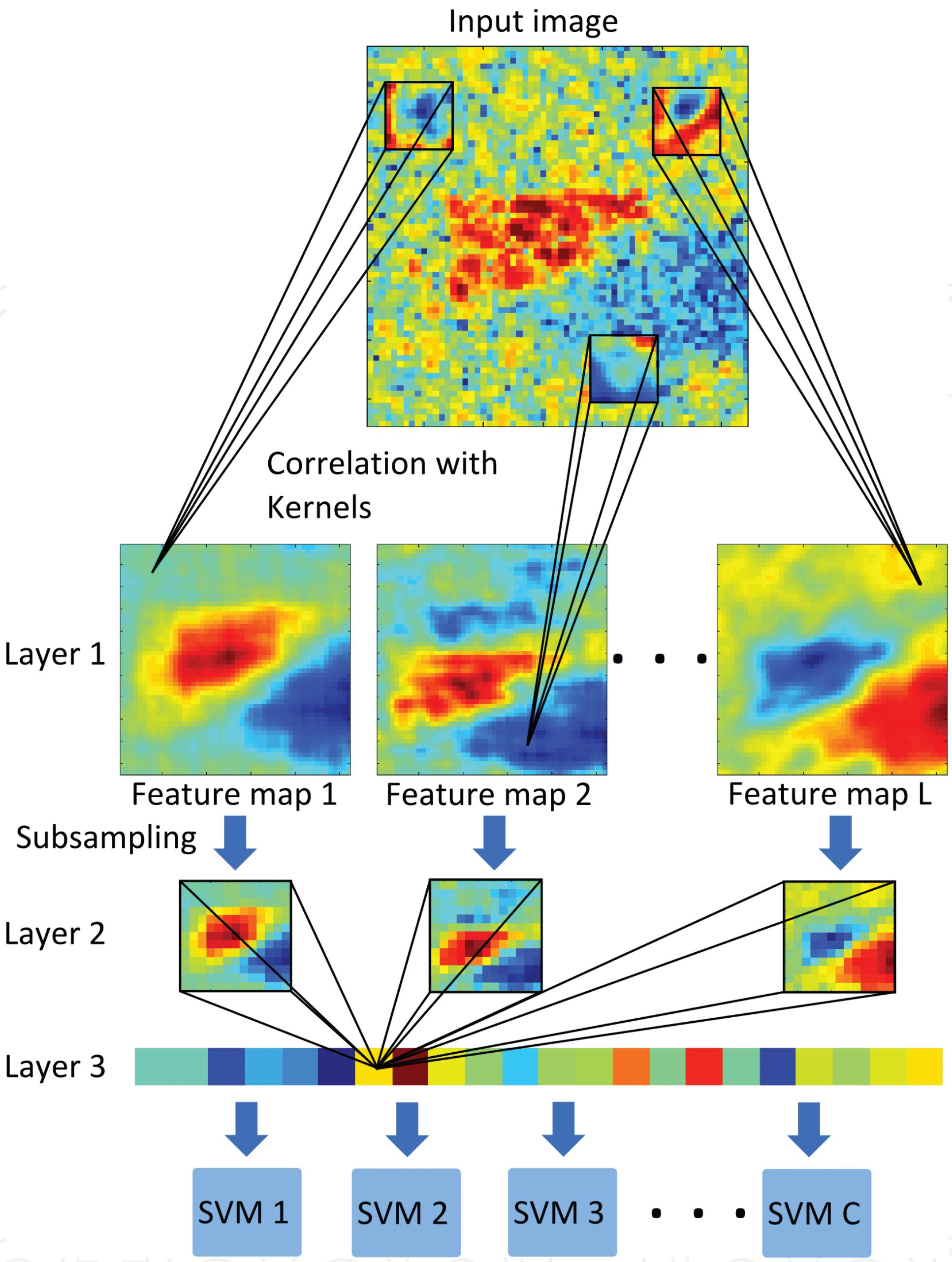


Figure 9. Structure of the used combination of CNN and SVMs.

Classifier	P_{CC}	P_{err}
Original CNN	96.00%	4.00%
CNN feature extraction and polynomial SVM	98.19%	1.81%
CNN feature extraction and RBF SVM	98.28%	1.72%

Table 1. Forced decision results of MSTAR dataset.

The results shown here are so called *forced decision* results, meaning that all images are classified by the highest output value, no rejection criteria like a certain confidence measure that has to be overcome is used. This and more results with the proposed classifier can be found in Ref. [38].

5. Rule-based-layer

Based on the abstracted situational picture derived by signal-processing and machine learning techniques, the cognitive radar system has to react to the perceived scene. Below, we illustrate a MDP based scheme to execute a-priori known, optimal illumination policies. In multifunctional radars, a radar resource manager has to schedule the individual illuminations into a serial radar timeline.

5.1. Optimal illumination policy

Markov-decision-processes (MDPs) are widely used in robotics to derive optimal control policies in stochastic environments. An agent in state s_i can execute different actions a_i, which with probability p_{ij} lead to a follow-up state s_j and a reward of r_{ij}. Different approaches, such as value-iteration or reinforcement-learning are used to determine an optimal policy $\pi = (s_i | a_i, s_j | a_j, \dots)$. The policy assigns to each state s_i an optimal action a_i which maximizes the expected reward. MDPs are well suited to model the perception-action-cycle of a radar, e.g. for tracking applications [39]. In the following, we illustrate an example for multi-stage classification from Ref. [40].

Three classes of targets $K = \{1, 2, 3\}$ can appear in a scenario with a priori-probability $\pi_1 = 0.1$, $\pi_2 = 0.2$ and $\pi_3 = 0.7$ (**Figure 10**). A low- and a high-resolution radar-mode ($mode = 1 | 2$) are available for up to five consecutive illuminations $t = \{1, 2, 3, 4, 5\}$, which are fused to a final declaration V using the Bayes rule (**Figure 11**). The policy describes the optimal illumination strategy with respect to the highest expectation for correctly classifying targets of class 1 ($V = 1 \Leftrightarrow$ Class 1, $V = 2 \Leftrightarrow \neg$ Class 1). A negative reward (cost) of 1 unit is assigned for a false alarm and 2 units for a missed detection.

The resulting multi-stage illumination policy is shown in **Figure 12**. Initially the target is illuminated with mode 2 and classified. Depending on the result $Y = 1, 2$, the strategy branches and finishes with a final declaration $V = 1, 2$. In a simulation of 100,000 Monte-Carlo runs, the static application of mode 1 resulted in accumulated costs of 20,000 (class 1 never detected, i.e. all missed detections). When randomly switching between mode 1 and 2, costs of 9063 occurred as opposed to the lowest cost of 4797 when using the optimal strategy.

5.2. Radar resource management

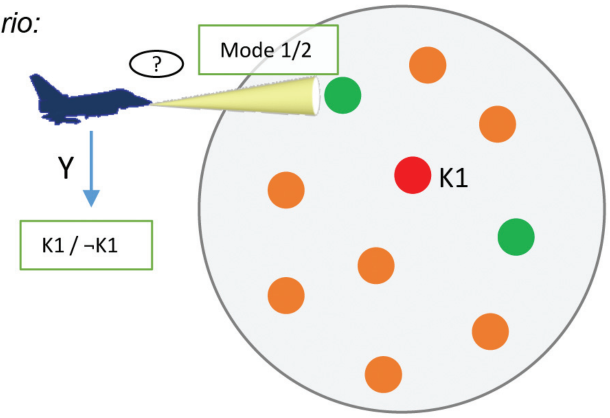
The illumination-strategy in **Figure 12** requires up to five consecutive illuminations of a target. As indicated in **Figure 1**, a multifunctional radar must simultaneously carry out additional tasks, in particular search for the new targets and track known targets. Since a shared aperture is used, the radar resource-manager schedules the radar timeline in time-multiplexing mode.

In the following, we simulate an airspace-surveillance radar rotating at 180°/s with electronic beam-steering.

5.2.1. Surveillance

The airspace is discretised depending on the beam width. Let $B_\varphi, B_\theta \in (0, 2\pi)$ be the azimuth and elevation opening angle respectively. The dwell time $\tau = 2\frac{r}{c}$ of an airspace section is

Scenario:



Confusion-matrix: $p(y|x = k, mode)$

Mode 1	K = 1	K = 2	K = 3
Y = K1	0.9	0.9	0.1
Y = ¬ K1	0.1	0.1	0.9

Mode 2	K = 1	K = 2	K = 3
Y = K1	0.75	0.15	0.1
Y = ¬ K1	0.25	0.85	0.9

Cost-matrix: $c(V|x = k)$

Kosten	K = 1	K = 2	K = 3
V = K1	0 *(cc)*	1 *(fa)*	1 *(fa)*
V = ¬ K1	2 *(md)*	0 *(cc)*	0 *(cc)*

Figure 10. Scenario, confusion- and cost matrix for the classification problem according to Ref. [40].

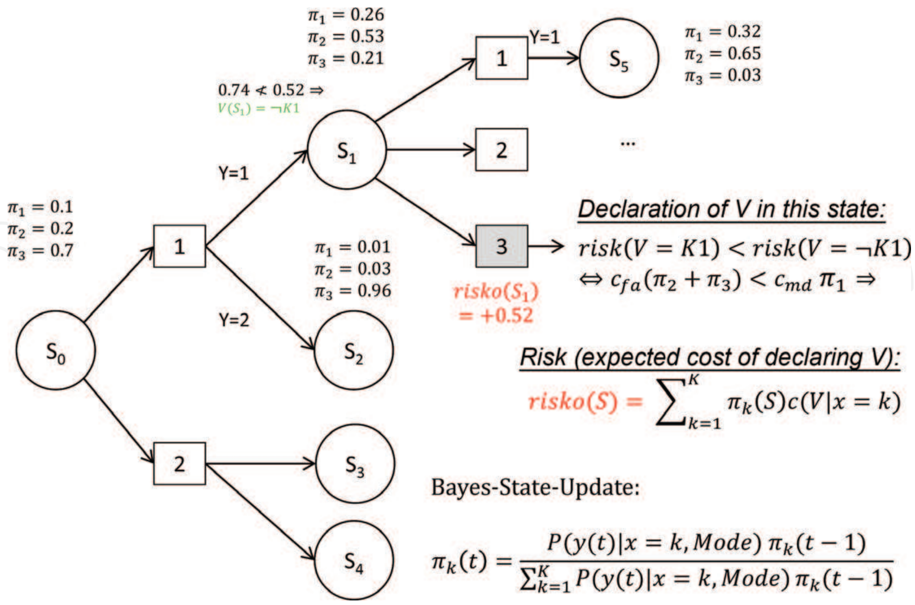

$\pi_1 = 0.26$
$\pi_2 = 0.53$
$\pi_3 = 0.21$
$0.74 \nless 0.52 \Rightarrow$
$V(S_1) = \neg K1$
Y=1
1
S$_5$
$\pi_1 = 0.32$
$\pi_2 = 0.65$
$\pi_3 = 0.03$
S$_1$
2
...
$\pi_1 = 0.1$
$\pi_2 = 0.2$
$\pi_3 = 0.7$
Y=1
1
$\pi_1 = 0.01$
$\pi_2 = 0.03$
$\pi_3 = 0.96$
3
Y=2
S$_0$
S$_2$
2
S$_3$
S$_4$
$risko(S_1) = +0.52$
Declaration of V in this state:
$risk(V = K1) < risk(V = \neg K1)$
$\Leftrightarrow c_{fa}(\pi_2 + \pi_3) < c_{md}\,\pi_1 \Rightarrow$
Risk (expected cost of declaring V):
$risko(S) = \sum_{k=1}^{K} \pi_k(S)c(V|x = k)$
Bayes-State-Update:
$\pi_k(t) = \frac{P(y(t)|x = k, Mode)\,\pi_k(t-1)}{\sum_{k=1}^{K} P(y(t)|x = k, Mode)\,\pi_k(t-1)}$

Figure 11. State-space, fusion, and selection of action (further measurement or final declaration V) to minimize the expected costs.

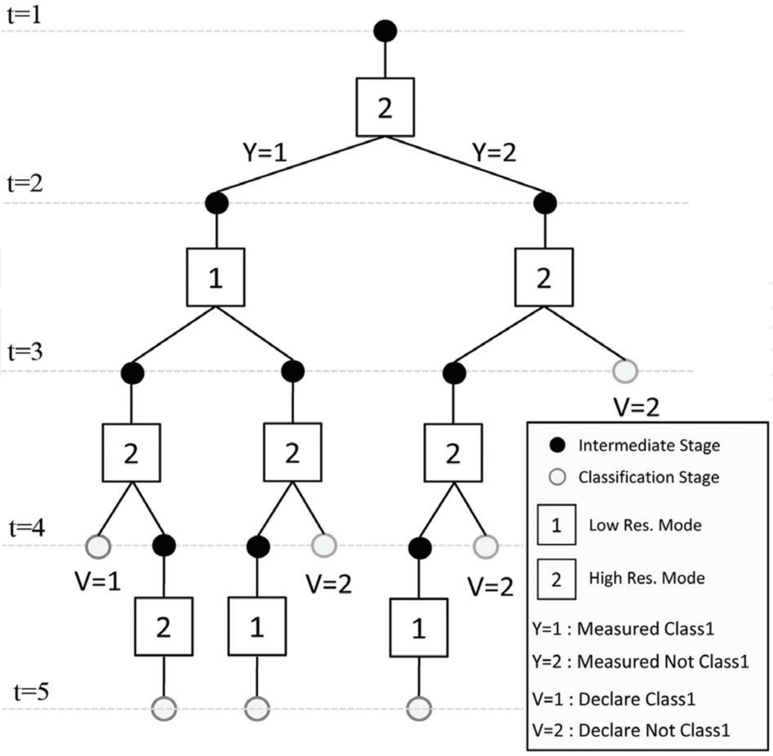


Figure 12. Optimal policy to the MDP.

chosen dependent on the range r of the target to guarantee that the whole range can be scanned within one transmit-receive process. Therefore the discretisation is only made in direction of azimuth and elevation. Since the transmit power decreases with increasing distance to the main lobe the borders are defined overlapping, i.e. a constant $d \in (0, 1)$ is selected for the discretisation (typical values are $d = 0.5$ or $d = 0.75$). If the maximum observable range and altitude are limited by R and H respectively, the airspace to be observed can be written as

$$\mathscr{L} = \left\{ x \in \mathbb{R}^3 | \|x\|_2 \le R \wedge h^\perp(x) \le H \right\}, \qquad R, H \in \mathbb{R}_+ \tag{26}$$

where the sensor is located in center of the coordinate system and $h^\perp(x)$ denotes the height of the target perpendicular to earth's surface. Then, after proper transformation the discretisation of $\mathscr{L}$ is given by

$$L_{ij} = \left(\frac{idB_\phi}{\cos\left(jdB_\theta\right)}, jdB_\theta \right) \in [0, 2\pi) \times \left[0, \frac{\pi}{2}\right]. \tag{27}$$

Here the factor $\cos(jdB_\theta)^{-1}$ compensates the circumstance that the same area (in steradians) engages a wider azimuth coverage on higher elevation than it does on lower elevation. When a surveillance task (see Section 5.2.3) is completed it is immediately regenerated with the desired revisit time to guarantee regular observation of the entire airspace.

5.2.2. Tracking

To be able to estimate the position of a target continuously in time all radar detections of a target T_i are put together into a track $\tilde{T}_i$. This is done by bringing them into physical relation using predefined dynamic models. A simple dynamic model assumes for example (statistically zero-mean) constant velocity which is variable through the (process-)noise in acceleration. To be able to determine which measurement belongs to which track the data association is done using scoring and global nearest neighbor approach (GNN) as it is described in Ref. [41]. In this case all unassociated detections generate a new track that applies as verified when the score exceeds a given threshold. In general a track is an estimation of the movement of the target, it contains information about the dynamic model, the covariance matrix P_i and an estimation $\widehat{x}_i(t)$ of the real state $x_i(t) = (p_i(t), v_i(t), \ldots)^T$ consisting of position $p_i(t)$, velocity $v_i(t)$ and for example acceleration $a_i(t)$ at time t. All tracks generated by the radar yield an estimation of the airspace situation (see **Figure 13**).

A more complex dynamic model was introduced by Singer [42]. The state $x(t)$ at time t can then shortly be written as $x(t) = (p(t), v(t), a(t))^T = (p(t), \dot{p}(t), \ddot{p}(t))^T$. The acceleration in this model is given by an ordinary differential equation

$$\begin{aligned} & p(t) = -\alpha\ddot{p}(t) + w(t) \\ \Leftrightarrow \quad & \dot{a}(t) = -\alpha a(t) + w(t) \end{aligned} \tag{28}$$

where α is the reciprocal of the maneuver time constant and $w(t)$ is Gaussian white noise. From Eq. (28) a discrete form of the Singer model at the k-th time step can be derived

$$x_k = F_k x_{k-1} + w_k \tag{29}$$

with discrete white noise w_k and process matrix F_k of the following form

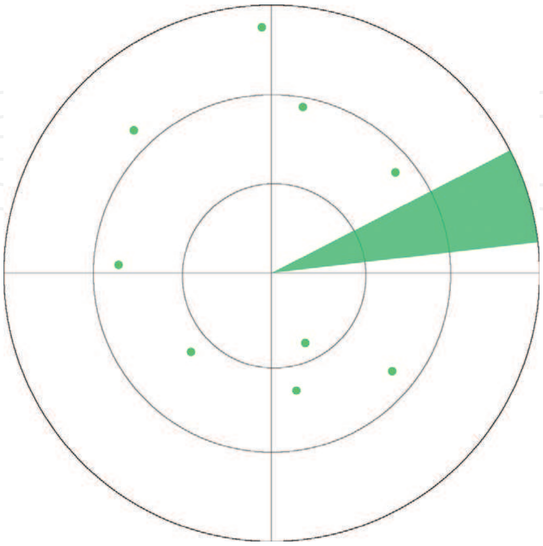


Figure 13. Airspace situation.

$$F_k = \begin{bmatrix} 1 & \Delta t & \frac{1}{\alpha^2}\left(-1+\alpha\Delta t + e^{-\alpha\Delta t}\right) \\ 0 & 1 & \frac{1}{\alpha}\left(1-e^{-\alpha\Delta t}\right) \\ 0 & 0 & e^{-\alpha\Delta t} \end{bmatrix} \tag{30}$$

where Δt denotes the time elapsed between time steps $k-1$ and k.

Recursive Bayesian estimators can be used to calculate the state x and the covariance matrix P (for a better readability the index i will be dropped from now on). One commonly used estimator is, for example, the (Extended) Kalman Filter (EKF) [41, 43]. In general the Kalman filter assumes a state transition model and an observation model

$$x_k = f(x_{k-1}) + w_k \tag{31}$$

$$z_k = h(x_k) + v_k \tag{32}$$

where z_k denotes the measurement, f and h are (not necessarily linear) functions, and w_k and v_k are additive, zero mean, white noises with process noise covariance Q_k and measurement noise covariance R_k respectively. In our case it is for example

$$h : \mathscr{L} \to [0, 2\pi)^2 \times \mathbb{R}, \begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} \varphi \\ \theta \\ r \end{pmatrix} = \begin{pmatrix} \operatorname{atan2}(y, x) \\ \arcsin\left(\frac{z}{r}\right) \\ \sqrt{x^2+y^2+z^2} \end{pmatrix} \tag{33}$$

the mapping between the state space $\mathscr{L}$ and the measurement in azimuth ϕ, elevation θ and range r. For the Singer model the state transition is a linear function with

$$f(x_{k-1}) = F_k x_{k-1}. \tag{34}$$

The EKF consists of two steps. First the state $x_{k\,|\,k-1}$ and the covariance $P_{k\,|\,k-1}$ are *predicted* using the previous information $x_{k-1\,|\,k-1}$ and $P_{k-1\,|\,k-1}$ (the index $k\,|\,k-1$ depicts the dependency of the estimates at time steps k and $k-1$):

$$\widehat{x}_{k|k-1} = f\left(x_{k-1|k-1}\right) \tag{35}$$

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1} \tag{36}$$

In the general case the matrix F_{k-1} is defined by

$$F_{k-1} = \frac{\partial}{\partial x} f\left(\widehat{x}_{k-1|k-1}\right). \tag{37}$$

Second the prediction will be *corrected* using the (erroneous) measurement z_k:

$$\widehat{x}_{k|k} = \widehat{x}_{k|k-1} + K_k\left[z_k - h\left(\widehat{x}_{k|k-1}\right)\right] \tag{38}$$

$$P_{k|k} = [I - K_k H_k] P_{k|k-1} \tag{39}$$

with observation matrix

$$H_k = \frac{\partial}{\partial x} h\left(\widehat{x}_{k|k-1}\right) \tag{40}$$

and Kalman gain

$$K_k = P_{k|k-1} {H_k}^T \left[H_k P_{k|k-1} {H_k}^T + R_k \right]^{-1}. \tag{41}$$

Process noise and measurement error accumulate over time until a new measurement is executed. This leads to a probability density of the track $\tilde{T}_i$ with state x_i.

The probability density is used to calculate the maximum time difference Δt that allows the track to stay in a predefined range relative accuracy:

$$\max \Delta t \tag{42}$$

subject to

$$\left\| \overline{P}^T P_i(t + \Delta t) \overline{P} \right\|_2 \leq \nu \sin\left(\min\left(B_\phi, B_\theta\right)\right) \left\| \tilde{p}_i(t) \right\|_2 \tag{43}$$

where $\overline{P}$ denotes a projection into the plane orthogonal to the beam direction and $\nu \in (0, 1)$ is the track sharpness. The time difference Δt is added to the tracking task of the track T_i and it is updated at every measurement.

5.2.3. Scheduler

In the simulation presented here a task is generated for each L_{ij} and placed into a sorted waiting queue (see **Figure 14**). The scheduler executes those tasks, whose time stamp do not lie in the past, in the given order. If tasks are delayed, they will be prioritized following the hierarchy of the waiting queue. The tasks inside the waiting queue are sorted according to their time stamps **Figure 15**.

5.2.4. Performance metrics

In this section three metrics are introduced to validate the performance of the resource manager.

One key element is the tracking accuracy. For the validation the distance between the estimated position $\tilde{p}_i(t)$ and the real position $p_i(t)$ of a target is calculated. The track does not contain any information about to which target it is related to, since the radar system does not know the ground truth. Therefore the track with the closest approach to the target is chosen as reference. The track sharpness is given as % of the beam width:

$$d_{TS} : \mathbb{R} \to \mathbb{R}, t \mapsto \max_i \min_j \frac{\left\| p_i(t) - \tilde{p}_j(t) \right\|_2}{\sin(\psi) \left\| p_i(t) \right\|_2}, \qquad \psi = \min\left(B_\phi, B_\theta\right) \tag{44}$$

The metric d_{TS} does not take into account whether the number of tracks matches the number of targets. Therefore the number of tracked targets $\#\tilde{T}$ is compared to the number of actually existing targets $\#T$ by the following metric:

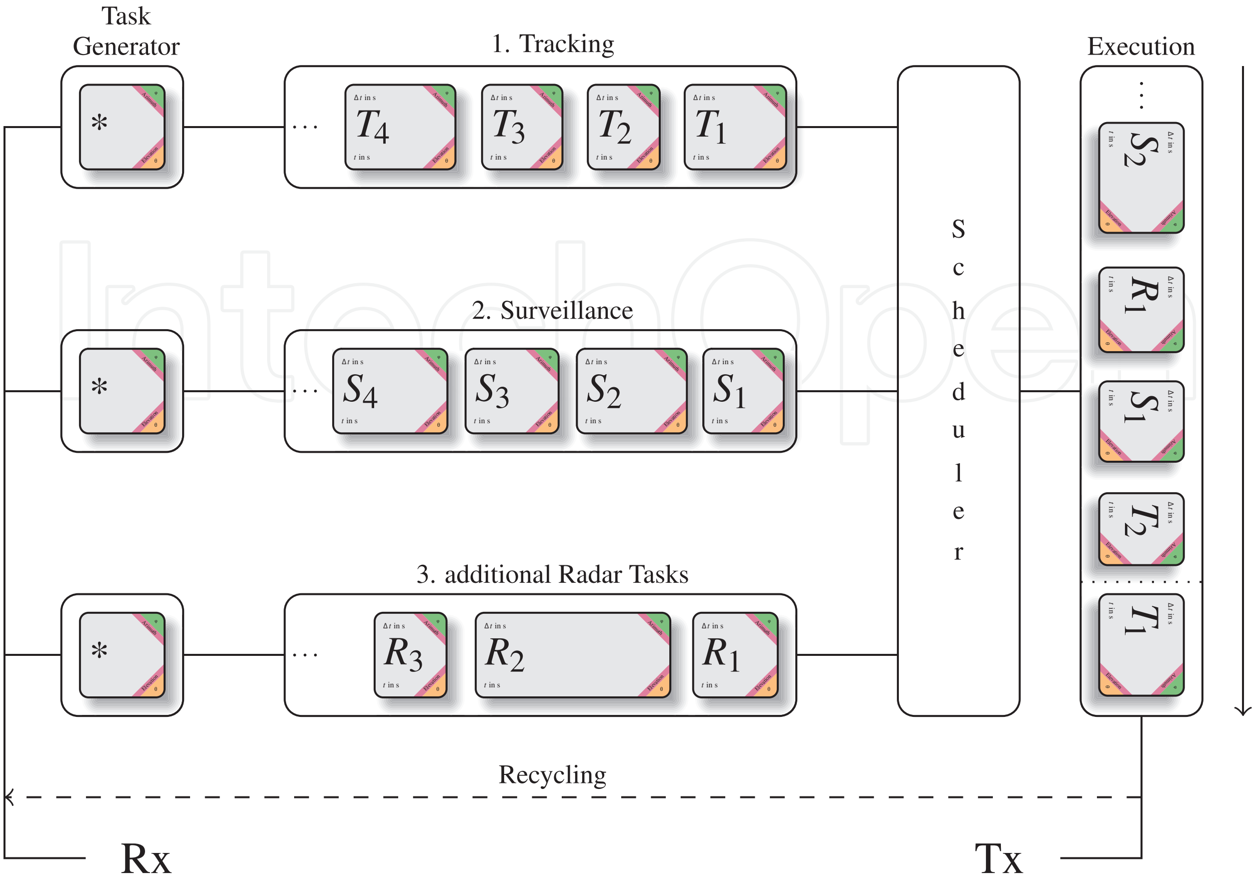


Figure 14. Schematic illustration of the scheduler.

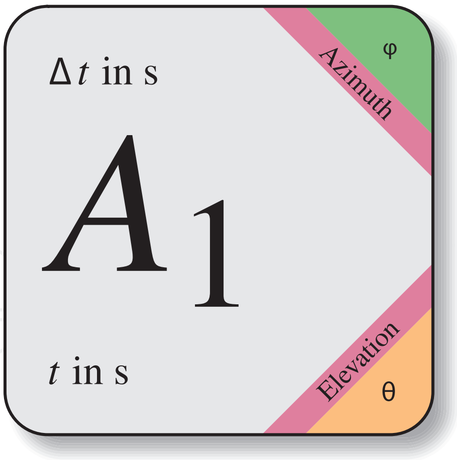


Figure 15. Detailed illustration of the task A_1. Containing a time stamp, the duration of the task, azimuth and elevation.

$$d_{\#} : \mathbb{R} \to \mathbb{R}, t \mapsto |\#T(t) - \#\tilde{T}(t)| \tag{45}$$

To evaluate the surveillance performance, the revisit time is considered. Let therefore be $t_{L_{ij}}$ the time of the last update of direction L_{ij} and let $\overline{L}(t)$ be the direction the radar is facing at time t. Then the metric is given by

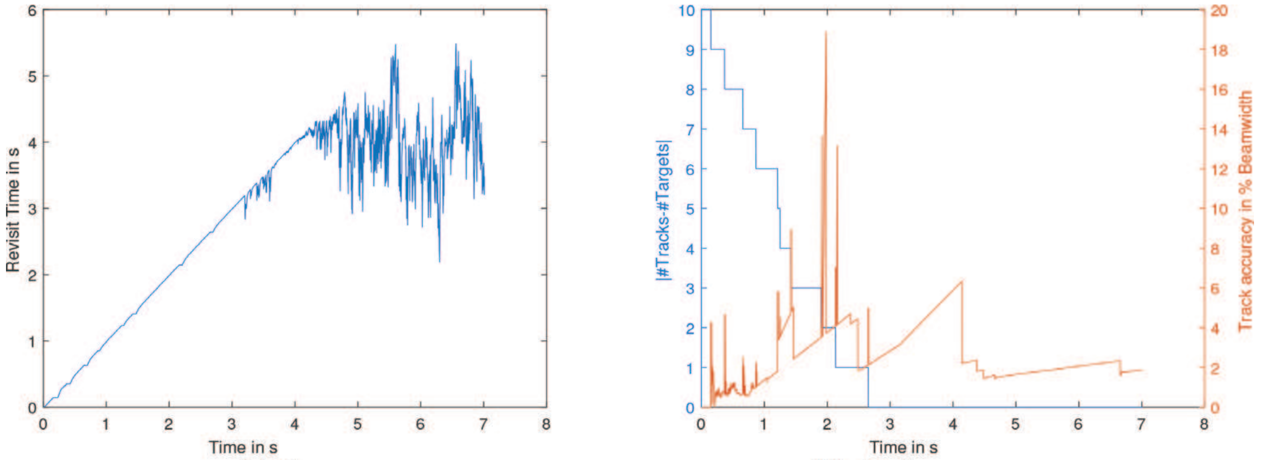
$$d_{RT} : \mathbb{R} \rightarrow \mathbb{R}, t \mapsto |t - t_{\overline{L}(t)}|. \tag{46}$$

5.2.5. Results

In this section the validation results of the simulation are presented. The actual airspace situation is depicted in **Figure 13**. **Figures 16** and **17** show the evaluations of the metrics defined in Section 5.2.4.

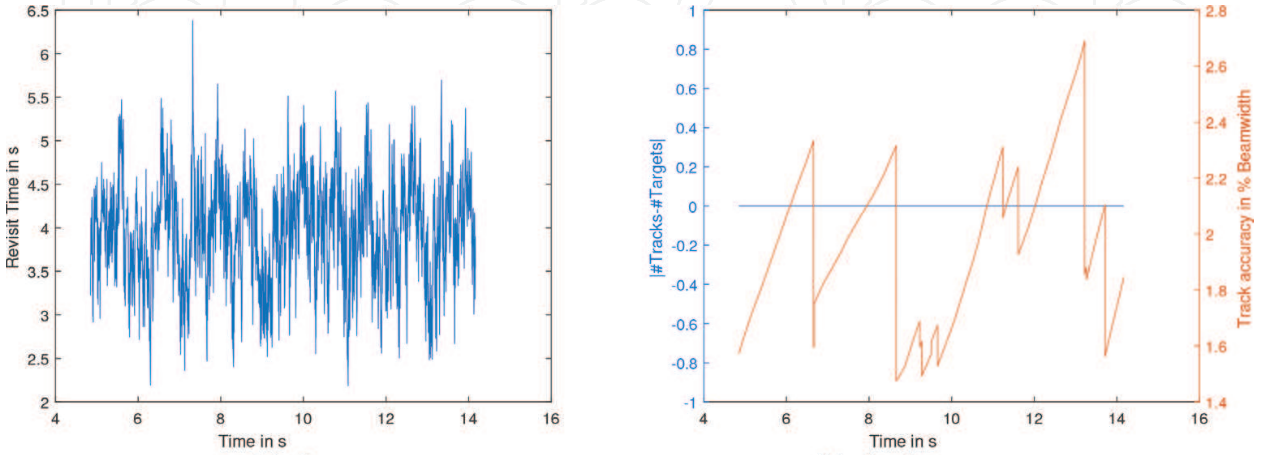
The simulation starts with an occupied airspace. This can be a difficult situation for the radar since pop-up targets significantly decrease the reaction time as the distance to the radar is shortened.

Figure 16(a) shows that the revisit time for the surveillance settles around a constant value after 4 seconds. The stepped line in **Figure 16(b)** shows that all targets are tracked in less than 3 seconds. The second line in (b) shows that the tracking accuracy is poor at the beginning of the simulation since the filters need several measurements to initialize correctly. **Figure 17** shows that the revisit time oscillates around 4 seconds and that the tracks are stable during routine operation.



(a) d_{RT} (b) $d_{\#}$; d_{TS}

Figure 16. Results during initialization.



(a) d_{RT} (b) $d_{\#}$; d_{TS}

Figure 17. Results during routine operation.

http://dx.doi.org/10.5772/intechopen.71365

6. Knowledge-based-layer

In this section, we discuss knowledge-based behavior of a cognitive radar. As discussed in Section 3, the knowledge-based layer works on structured a-priori knowledge about the application domain and its goals and constraints. Automated planning or optimisation tools can be applied to generate mission-level commands that, for example, control the trajectory of the sensor-carrying platform.

Below, we discuss an illustrative trajectory planning problem for a 6-DOF robotic manipulator arm that carries a UWB sensor able to work in synthetic aperture radar (SAR) mode. Results from a real measurement setup using a ST Robotics R17 robot arm are also shown. The sensor has one transmitter and one receiver in a typical common-offset arrangement (**Figure 18**).

6.1. Robot trajectory planning

The spatial resolution and processing gain that the system can achieve ultimately depend on the trajectory and velocity profile of the sensor head. The constraints can be modeled as an optimisation problem to obtain a feasible, collision-free trajectory of the end-effector of the manipulator arm in Cartesian coordinates that minimizes observation time.

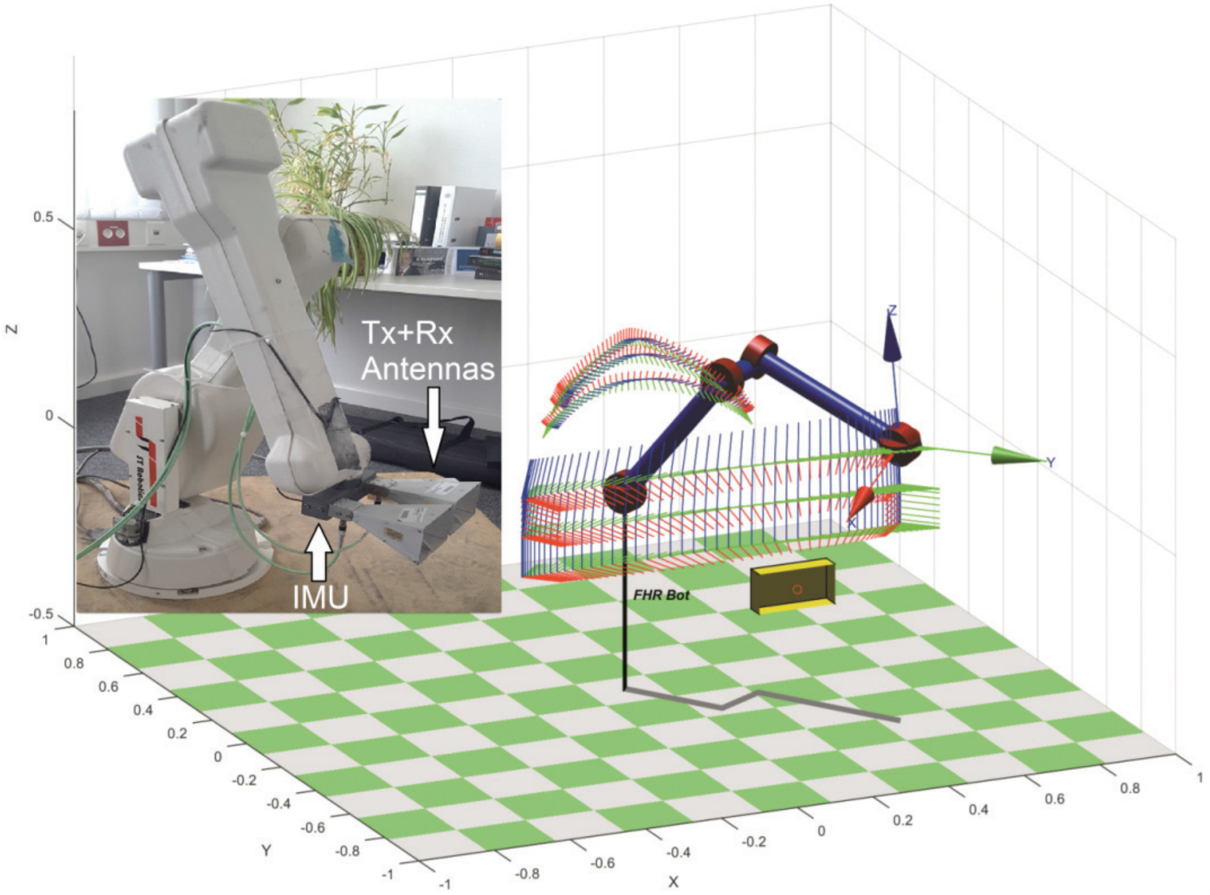


Figure 18. Trajectory planning for IED inspection with R17HS robot arm.

6.1.1. Sensor characteristics and trajectory constraints

The radar sensor under consideration uses a selectable center-frequency from 3 to 8 GHz and 4 GHz of bandwidth, resulting in 3.75 cm of range resolution. The center frequency can be tuned according to a particular target or propagation environment (ground penetration, through-the-wall imaging, IED inspection…). The horn-type antennas can be rotated to exploit polarization diversity. The sensor is able to operate in stripmap or spotlight SAR modes using linear trajectories. Several parallel trajectories can be combined for 3D imaging. The mobility of the arm could be further exploited to generate non-linear trajectories around a target to obtain a more accurate 3D reconstruction.

In order to obtain a similar resolution in cross-range than in range the trajectory planning must (aim to create at least an aperture of 0.5 to 1.3 times the distance to the target in both dimensions (azimuth and elevation) depending on the center frequency used by the system 3 to 8 GHz respectively).

High resolution imaging can only be achieved with an even higher precision positioning. The 3D-trajectory of the sensor needs to be measured and synchronized with the sensor data. For that purpose, accelerometers and gyroscopes from an attached inertial measurement unit (IMU) are used. The IMU drift is additionally stabilized using the hardware readout of optical encoders of the robot arm joints controlled by step-motors.

Two other important parameters to be considered for the trajectory planning are the optimal size of the scanning area and the sampling requirements. Considering the case of planar acquisition geometries working in stripmap mode, to obtain full resolution imaging of the total area of interest, an additional half beam aperture must be extended in both dimensions.

Another important parameter is related with the sampling requirements of a particular acquisition. The measurement positions in the synthetic radar aperture require a minimum spacing

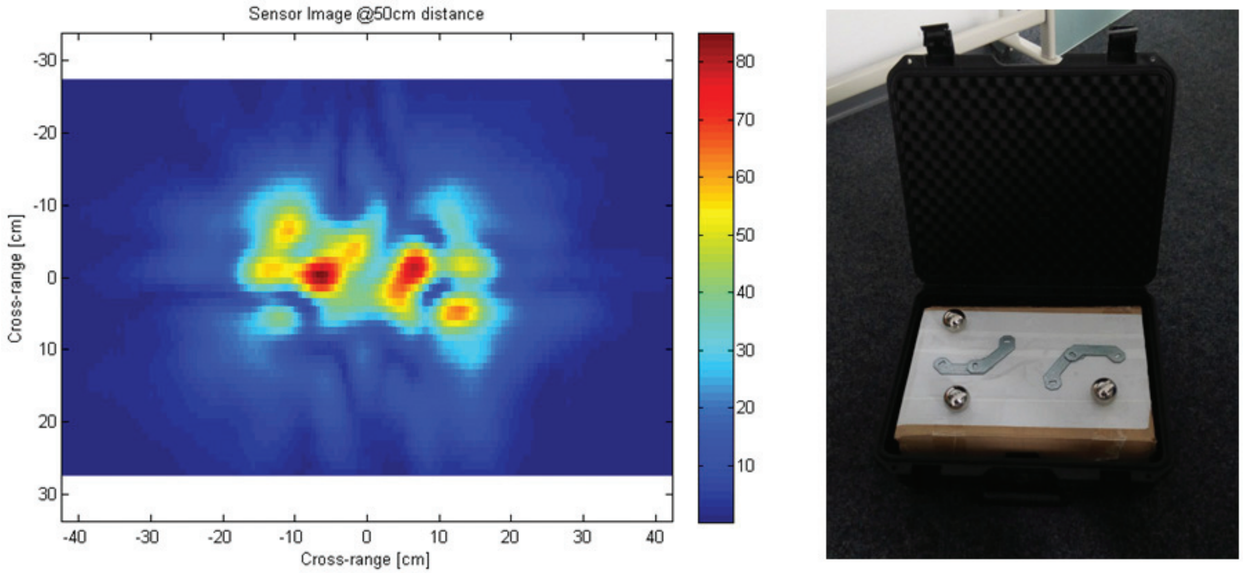


Figure 19. Image of objects inside a suitcase using the robot arm.

in order to sample adequately the phase history associated with all the scatterers. If the distance between measurements is too large the Nyquist criterion is not fulfilled and artifacts may appear in the reconstructed image.

It must be considered also that signal propagation in dielectric materials (ground, wall) will shrink the wavelengths, and sampling requirements become then even more stringent [44]. A previous estimation of the dielectric permittivity of the propagation media may further optimize the acquisition geometry and the imaging process. **Figure 19** shows an example of an image obtained with the robot arm using some reference objects inside a plastic suitcase. The trajectory followed by the sensor has been planned considering the constraints previously mentioned to obtain unaliased high-resolution images of the total area of interest.

7. Conclusions

In this article, a three-layered cognitive radar-architecture based on the Rasmussen model was presented. Several examples illustrated technologies to implement the cognitive subfunctions in a radar system.

For the skill-based layer, an approach for matching a waveform to the target transfer function was shown. In addition, spectrum sensing methods can be used to adapt the transmit signal to the electromagnetic environment. Rule-based behavior can be implemented using Markov-decision processes (MDPs) to compute optimal illumination policies. For a shared-aperture multifunctional radar, radar-resource management approaches are required to schedule the radar timeline. For knowledge-based behavior, an example for sensor-controlled trajectory generation of a robotic-arm were presented.

The different layers of the architecture encompass a broad range of time-scales and levels of abstraction. The full potential is achieved, if all layers interact consistently. This and further experimental validation of the approach are currently investigated at FHR.

Author details

Stefan Brüggenwirth*, Marcel Warnke, Christian Bräu, Simon Wagner, Tobias Müller, Pascal Marquardt and Fernando Rial

*Address all correspondence to: stefan.brueggenwirth@fhr.fraunhofer.de

Fraunhofer-Institut für Hochfrequenzphysik und Radartechnik FHR, Wachtberg, Deutschland

References

[1] Haykin S. Cognitive radar: A way of the future. Signal Processing Magazine. USA: IEEE. Jan. 2006;**23**(1):30-40

[2] Guerci JR. Cognitive Radar: The Knowledge-Aided Fully Adaptive Approach. Artech House, Incorporated, 2010. ISBN: 9781596933651

[3] Ender J, Brüggenwirth S. Cognitive radar-enabling techniques for next generation radar systems. In: 2015 16th International Radar Symposium (IRS). IEEE. 2015. pp. 3-12

[4] Rasmussen J. Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. Systems, Man and Cybernetics, IEEE Transactions on SMC. May 1983;**13**(3):257-266

[5] Schulte A. Kognitive und kooperative Automation zur Führung unbemannter Luftfahrzeuge. Online Journal Kognitive Systeme. 2013;**1**

[6] Sheridan TB. Telerobotics, Automation, and Human Supervisory Control. Cambridge MA: MIT Press; 1992

[7] Wohler M, Schulte A. On board assistant system for UAS integration into civil airspace. In: AIAA Infotech@ Aerospace (I@ A) Conference. 2013. p. 5047

[8] Onken R, Schulte A. System-Ergonomic Design of Cognitive Automation. Heidelberg: Springer; 2010

[9] Putzer H, Onken R. COSA–A generic cognitive system architecture based on a cognitive model of human behavior. Cognition, Technology & Work. 2003;**5**(2):140-151

[10] Brüggenwirth S, Schulte A. COSA2 cognitive system architecture with centralized ontology and specific algorithms. In: 2012 International Conference on Systems, Man, and Cybernetics (IEEE SMC 2012). IEEE. 2012

[11] Gjessing DT. Environmental remote sensing, part 1: Methods based on scattering and diffraction of radio waves. Physics in Technology. 1979;**10**:266-271

[12] Bell MR. Information theory and radar waveform design. Proceedings. IEEE International Symposium on Information Theory. Jan 1993:428-428. DOI: 10.1109/ISIT.1993. 748744

[13] Schreier PJ, Scharf LL. Statistical Signal Processing of Complex-Valued Data: The Theory of Improper and Noncircular Signals. Cambridge: Cambridge University Press; 2010

[14] Wysocki T, Razavi H, Honary B. Digital Signal Processing for Communication Systems. New York: Springer US; 1997 9780792399322

[15] An Assessment of the Near Term Viability of Accommodating Wireless Broadband Systems in the 1675–1710 MHz, 1755, 1780 MHz, 3500–3650 MHz, and 4200–4220 MHz, 4380–4400 MHz Bands. Tech. rep. U.S. Department of Commerce; 2010

[16] Use of the Bands 5150–5250 MHz, 5250–5350 MHz and 5470–5725 MHz by the Mobile Service for the Implementation of Wireless Access Systems Including Radio Local Area Networks. Tech. rep. ITU-R Resolution 229; 2012

[17] Axell E, et al. Spectrum sensing for cognitive radio. IEEE Signal Processing Magazine. 2012;**29**(3):101-116

[18] Yucek T, Arslan H. A survey of spectrum sensing algorithms for cognitive radio applications. IEEE Communications Surveys & Tutorials. 2009;**11**(1):116-130

[19] Tandra R, Sahai A. SNR walls for signal detection. IEEE Journal of Selected Topics in Signal Processing. 2008;**2**(1):4-17. DOI: 10.1109/JSTSP.2007.914879

[20] Gardner W. Measurement of spectral correlation. IEEE Transactions on Acoustics, Speech, and Signal Processing. 1986;**34**(5):1111-1123

[21] Dandawate AV, Giannakis GB. Statistical tests for presence of cyclostationarity. IEEE Transactions on Signal Processing. 1994;**42**(9):2355-2369

[22] Zeng Y, Liang Y-c. Eigenvalue-based Spectrum sensing algorithms for cognitive radio. IEEE Transactions on Communications. 2009;**57**(6):1784-1793

[23] Stinco P, Greco MS, Gini F. Spectrum sensing and sharing for cognitive radars. IET Radar, Sonar & Navigation. 2016;**10**(3):595-602

[24] Kovarskiy JA, et al. Comparison of RF spectrum prediction methods for dynamic spectrum access. In: KI Ranney and A Doerry, ed. Proc. SPIE 10188. May 2017, p. 1018819. DOI: 10.1117/12.2262306

[25] Tian Z, Giannakis GB. A wavelet approach to wideband spectrum sensing for cognitive radios. In: 2006 1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications. June 2006. pp. 1-5. DOI: 10.1109/CROWNCOM.2006.363459

[26] Martone A, et al. Passive sensing for adaptable radar bandwidth. In: 2015 IEEE Radar Conference (RadarCon). May 2015. pp. 0280-0285. DOI: 10.1109/RADAR.2015.7131010

[27] Kirk BH, et al. Cognitive software defined radar: Waveform design for clutter and interference suppression. In: Ranney KI, Doerry A, editors. SPIE Defense + Security. SPIE Proceedings. SPIE. 2017. p. 1018818

[28] Bengio Y. Learning deep architectures for AI. Foundations and Trends in Machine Learning. 2009;**2**(1):1-127. ISSN: 1935-8237. DOI: 10.1561/2200000006

[29] Haykin S. Neural Networks and Learning Machines. 3rd ed. Upper Saddle River, New Jersey: Pearson Education Inc.; 2009

[30] Zhao Q, et al. Synthetic aperture radar automatic target recognition with three strategies of learning and representation. Optical Engineering. 2000;**39**(5):1230-1244. DOI: 10.1117/1.602495

[31] leCun Y. Generalization and Network Design Strategies. Technical Report CRG-TR-89-4. Department of Computer Science. Toronto, Canada: University of Toronto; June 1989

[32] LeCun Y, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE. Nov. 1998;**86**(11):2278-2324. ISSN: 0018-9219. DOI: 10.1109/5.726791

[33] Schachter BJ. Target classification strategies. Proceedings of SPIE. 2015;**9476**:947602-947602. DOI: 10.1117/12.2176193

[34] Lauer F, Bloch G. Incorporating prior knowledge in support vector machines for classification: A review. Neurocomputing. 2008;**71**(7–9):1578-1594 ISSN: 0925-2312

[35] Diemunsch JR, Wissinger J. Moving and stationary target acquisition and recognition (MSTAR) model-based automatic target recognition: Search technology for a robust ATR. Proceedings of SPIE. 1998;**3370**:481-492. DOI: 10.1117/12.321851

[36] Wagner S, Barth K, Brüggenwirth S. A deep learning SAR ATR system using regularization and prioritized classes. In: 2017 IEEE Radar Conference (RadarConf). May 2017. pp. 0772-0777. DOI: 10.1109/RADAR.2017.7944307

[37] Vapnik VN. The Nature of Statistical Learning Theory. Second. Statistics for Engineering and Information Science. New York: Springer-Verlag; 2000

[38] Wagner SA. SAR ATR by a combination of convolutional neural network and support vector machines. IEEE Transactions on Aerospace and Electronic Systems. Dec. 2016;**52** (6):2861-2872. DOI: 10.1109/TAES.2016.160061

[39] Haykin S. Cognitive Dynamic Systems: Perception-Action Cycle, Radar and Radio. Cambridge: Cambridge University Press; 2012. 9780521114363

[40] Castanon DA. Approximate dynamic programming for sensor management. In: Decision and Control, 1997. Proceedings of the 36th IEEE Conference on. Vol. 2. IEEE. 1997. pp. 1202-1207

[41] Blackman S, Popoli R. Design and Analysis of Modern Tracking Systems. Norwood, MA: Artech House; 1999

[42] Singer RA. Estimating optimal tracking filter performance for manned maneuvering targets. IEEE Transactions on Aerospace and Electronic Systems. 1970;**4**:473-483

[43] Van Trees HL. Detection, Estimation, and Modulation Theory. New York, NY: John Wiley & Sons; 2004

[44] Grasmueck M, Weger R, Horstmeye H. Full resolution 3-D GPR imaging. Geophysics. 2005;**70**(1):12-19