Filter Design Explorer 1.0 User's Guide

Web: www.cadescenter.com

Email: info@cadescenter.com

Contents

1. INTRODUCTION 3

1.1 Key Features of FDE 1.0 3

1.2 Optimization Problems Solved by FDE 1.0 4

1.3 Downloading, Installation and Licensing 5

2. OPTIMIZATION CONCEPTS 6

2.1 Design Variables 6

2.2 Range of Design Variables 6

2.3 Objective Function 7

2.4 Constraints 7

2.5 Single Objective Optimization 7

2.7 Constrained Optimization 8

2.8 Penalty Functions 8

3. OPTIMIZATION METHOD IN FDE 1.0 9

3.1 Surrogate Model Assisted Differential Evolution Algorithm for Filter Optimization (SMEAFO) .. 9

3.4 Parameter Settings for SMEAFO 10

4. BASIC USE OF FDE 1.0 13

4.1 Using the Software 13

4.2 Essential Tutorials 23

5.0 PROBLEM EXAMPLES 27

5.1 WR-3 Band Waveguide Filter 27

REFERENCES 30

APPENDIX 1.0 : FDE 1.0 INSTALLATION QUICK GUIDE 31

1. INTRODUCTION

Filter Design Explorer (FDE) 1.0 is a software tool with a Graphical User Interface (GUI) aiming to optimize filter geometrical design parameters satisfying design specifications defined by the user from a given structure.

The user's inputs include: (1) a filter design evaluation file [in 1.0, this can be a CST Microwave Studio model or a user defined MATLAB function (e.g., MATLAB antenna toolbox-based evaluation, user defined MATLAB functions invoking other 3D EM simulators, etc.)], (2) The design variables and their ranges, (3) The optimization goal and constraint(s). In the FDE 1.0 platform, three types of optimization are supported *(see Section 1.2 for more details)*. **The output** is the designed filter layout with optimal geometrical design parameters *(see sections 4 and 5 for more details)*.

1.1 Key Features of FDE 1.0

Microwave filter design is very challenging. Filters are typically narrow band devices, so the desired design solution always lies in a very narrow valley of the solution space. Moreover, the filter design landscape often has numerous local optima. Carrying out global optimization often costs considerable time (much longer than optimizing antennas), while carrying out local optimization may often be trapped in a local optimum. Hence, a popular way is to firstly generate an initial design and then perform optimization based on it. Fortunately, there are several methods to obtain an initial design, among which, the widely used ones are equivalent circuits and coupling matrix. They will be included in future FDE tools.

However, the quality of the initial design can vary. In some cases, the initial design is of high quality, showing good responses. In such cases, performing local optimization is recommended and FDE 1.0 is NOT designed for this purpose. Optimizers in CST, HFSS and MATLAB, as well as space mapping methods, are recommended to be used. In some other cases, the initial design is of low quality, showing poor responses. Thus, both local and global optimization methods have difficulty in addressing them. FDE 1.0 is designed for this purpose. Some of the key features of FDE 1.0 include, (but are not limited to):

- **Efficient Global Optimization:**
  FDE 1.0 implements the SMEAFO algorithm in [1]. SMEAFO is a surrogate model-assisted hybrid (combining global and local optimization) method specially designed for optimizing filters. It has been shown that the SMEAFO method obtains a 20 – 30 times speed improvement compared to standard global optimization methods, while getting comparable (or even better) results [1]. As such, the adoption of FDE 1.0 for your design exploration and optimization almost guarantees a high-quality design with highly reduced computational effort (decreasing 3 months' optimization time to 2 – 3 days).

- **Intensive Usability Consideration:**
  FDE 1.0 has been designed to adequately meet the usability requirements of engineers and researchers with expertise in filter analysis and design, but without deep knowledge of optimization. FDE 1.0 is a one-button optimization tool. For normal cases optimizing the S-parameters, the user does not need to program anything. Instead, by simply filling in appropriate information through the GUI, the optimization problem can be set. FDE 1.0 also remains flexible. For special cases, user-defined functions can be used to define most

responses that the user is interested in. In addition, FDE 1.0 provides a seamless link with CST Microwave Studio.

Another critical issue is that many filter designers struggle with the setting of parameters for optimization algorithms. Though these parameters affect the performance of the optimization algorithms, more often than not, there are no clear rules on how to set them. FDE 1.0 addresses this problem by automating the parameter settings for its algorithms in instances where the user opts not to specify the parameter settings. With the exception of a single parameter (the setting rule is provided in section 3), the appropriate parameters are adaptively calculated and set as the default within FDE 1.0 when the user opts for the automatic settings. Of course, these automatic parameters are based on real-world filter design testing. *(See section 3 for more details on parameter settings.)*

- **Full Tutorial on Filter Design Optimization including Many Useful Tips:**
  As a complementary guide for FDE 1.0, full tutorials covering many useful tips are provided. FDE 1.0 also supports the use of these tips through its GUI. By studying the tutorials (including videos), filter designers without much optimization expertise can quickly handle effective filter optimization problems in a much easier and effective way.

1.2 Optimization Problems Solved by FDE 1.0

FDE 1.0 solves three main types of optimization problems, illustrated in Fig. 1 with a 300-GHz waveguide filter example. These optimization problems are typical in filter design exploration and synthesis.

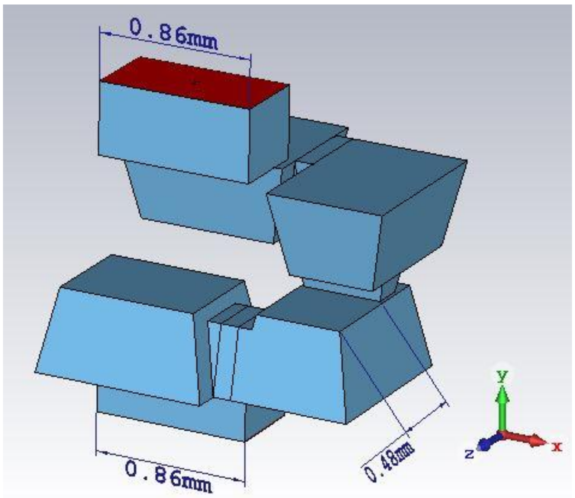

Fig. 1. 3-D Geometry of the WR-3 Band Waveguide Filter [1]

- ***Specification Satisfaction***
  Example 1: The filter designer wants to perform design exploration satisfying the following requirements:
  - $\max(|S_{11}|) \leq -20dB$
  - $8GHz; \quad Passband; \quad Centerd: 300GHz$

- $Stopbands;\quad 280GHz-292GHz;\quad 308GHz-320GHz$

- ***Goal Optimization***

  Example: The filter designer wants to perform design exploration minimizing $\max(|S_{11}|)$ between 296GHz and 304GHz.

- ***Constrained Optimization***

  Example: The filter designer wants to perform design exploration minimizing $\max(|S_{11}|)$ between 296GHz and 304GHz, subject to the following constraints:
  - $\min(|S_{11}|) \geq -1dB;\quad 280GHz-292GHz$
  - $\min(|S_{11}|) \geq -1dB;\quad 308GHz-320GHz$

See Section 2 for further details on these concepts.

1.3 Downloading, Installation and Licensing

FDE 1.0 is free software and can be downloaded from the CADES FDE website. The licensing should follow the policy of the Computer-aided Design and Engineering Software (CADES) Research Center (cadescenter.com). The CADES Research Center is an inter-university research and development center composed of partners mainly from Europe, North America and Asia. Several key issues of licensing are:

- FDE 1.0 is prohibited from cracking, copying, distribution, retail and any other commercial activity.
- For commercial use, the interested parties or organizations must contact the CADES Research Center to obtain the necessary permits (contacting CADES is available through the CADES homepage and the product pages).
- Each license expires in 90 days, but the user can download (updated) software from the CADES website (cadescenter.com) for free. This policy is to make sure that the user can keep up with the most recent version of FDE while providing feedback for evaluation and improvement.

FDE 1.0 runs in the MATLAB environment and is compatible with Windows and Linux (although not quite all versions) systems. MATLAB 2014 version or above is needed. The necessary toolboxes include the Optimization Toolbox, Machine Learning and Statistics Toolbox and Parallel Computing Toolbox. In FDE 1.0, a user-friendly seamless link with CST Microwave Studio is provided, which means FDE 1.0 can directly work with a CST model. However, CST Microwave Studio is not itself included in FDE 1.0. To use this function, the CST Microwave Studio version 2014 or above will be required. Though other 3D EM simulators are also supported on the FDE 1.0 platform, users will need to develop their own respective interfacing methods.

Installation of FDE 1.0 is extremely simple. A quick installation guide is provided in Appendix 1.0.

2. OPTIMIZATION CONCEPTS

This section introduces the basic concepts of optimization to the novice user. Users with existing expertise in optimization may jump to Section 3.

2.1 Design Variables

As the name suggests, design variables are parametric features in RF and EM engineering design whose metric values can be varied to improve the overall profile and quality of the design. With respect to antennas, filters and other RF or EM problems, design variables are usually measurable geometric or spatial components of the design. For instance, Fig.2 and Table 1 reveal the set design variables of a microstrip filter problem proposed for optimization [1].

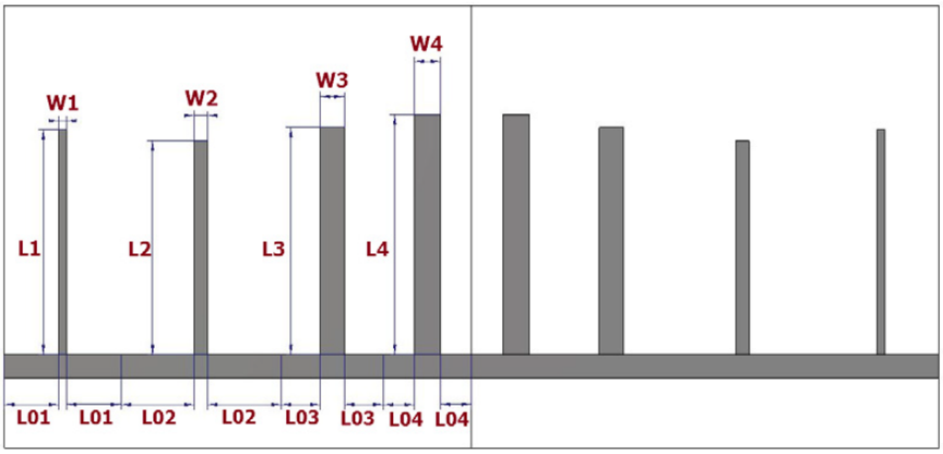

Fig. 2: Front View of a Microstrip Filter [1]

Table 1: Design Variables of a Microstrip Filter [1]

RANGE OF THE DESIGN VARIABLES (ALL SIZES IN MM) FOR MICROSTRIP FILTER DESIGN EXPLORATION												
Name of Variable	**W1**	**W2**	**W3**	**W4**	**LO1**	**LO2**	**LO3**	**LO4**	**L1**	**L2**	**L3**	**L4**
Lower Bound	0.5	0.5	0.5	0.5	2	2	2	2	7	7	7	7
Upper Bound	3.5	3.5	3.5	3.5	8	8	8	8	18	18	18	18

Combining Fig. 2 and Table 1, it can be seen that design variables constitute the parametric boundaries of the microstrip filter being examined.

2.2 Range of Design Variables

Extending the concept of design variables introduced earlier, the range of variables for an optimization problem can be simply defined as the lower and upper bounds specified for the design variables. In the context of antennas, filters and other RF or EM problems, the range of variables are usually the minima and maxima for the measurable geometrical or spatial components of the design. Table 1 shows the range of variables (minima and maxima) for the parametric features of a microstrip filter proposed for optimization [1]. We generally assume that the optimal solution is not located outside of any of the above ranges. It should be noted that filter problems are characterized by two sets of ranges due to the characteristics of the design landscape in FDE 1.0 *(see section 3.4 for more details)*.

2.3 Objective Function

In simple terms, an objective function is a mathematical equation that can model the objective(s) of an optimization problem and is applied along with any constraints [2]. Generally, objective functions depict optimization problems as minimization or maximization tasks. Equation 1, for example, depicts an objective function $f(x)$ for an optimization problem, "s.t" or 'subject to' constraints $g_1(x), g_2(x), g_3(x)$ and $g_4(x)$ [3]:

$$
\begin{aligned}
&\min\ f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7 - 10x_6 - 8x_7 \\
&s.t. \\
&\quad g_1(x) = 2x_1^3 + 3x_2^4 + x_3 + 4x_4^3 + 5x_5 - 127 \le 0 \\
&\quad g_2(x) = 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \le 0 \\
&\quad g_3(x) = 23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196 \le 0 \\
&\quad g_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0 \\
&For:\quad -10 \le x_i \le 10, i = 1,2,\ldots 7
\end{aligned}
\tag{1}
$$

2.4 Constraints

In optimization, a constraint is a condition that any solution to the optimization problem must satisfy. Though constraints can be specified mathematically as equalities or inequalities, they are usually depicted as inequalities in filter design optimization [2 – 4]. For instance, the objective function in equation 1 is subject to the constraints $g_1(x), g_2(x), g_3(x)$ and $g_4(x)$.

$$
\begin{aligned}
&g_1(x) = 2x_1^3 + 3x_2^4 + x_3 + 4x_4^3 + 5x_5 - 127 \le 0 \\
&g_2(x) = 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \le 0 \\
&g_3(x) = 23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196 \le 0 \\
&g_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0
\end{aligned}
\tag{2}
$$

By definition, an optimization problem subject to constraints can be called 'constrained optimization' and an optimization problem without constraints 'unconstrained optimization'.

2.5 Single Objective Optimization

From [2], an unconstrained single objective optimization problem can be described by equation 3:

$$
\begin{aligned}
&\min f(x) \\
&s.t. \\
&x \in [a,b]^d
\end{aligned}
\tag{3}
$$

where x is the vector of decision variables; d is the dimension of x; $[a,b]^d$ are the search ranges of the decision variable x, and $f(x)$ is the objective function. The optimal solution is the value of $x \in [a,b]^d$ such that there is no other point $x* \in [a,b]^d$ with $f(x*) < f(x')$ (assuming a minimization problem).

2.7 Constrained Optimization

A constrained optimization problem can be depicted using equation 4 as follows:

$$\begin{aligned} &\min f(x) \\ &s.t. \\ &g_i(x) \le 0, i = 1,2,...,k. \\ &x \in [a,b]^d \end{aligned} \tag{4}$$

where $f(x)$ depicts the optimization goal and $g_i(x)$ are the constraints. Constrained optimization can be single-objective or multi-objective. In filter optimization, we generally consider that feasible solutions (satisfying all the constraints) are better than infeasible solutions. For two feasible solutions, their optimality is determined by the objective function value or Pareto dominance (Pareto optimization is not available in FDE 1.0).

2.8 Penalty Functions

Often described in the context of constrained optimization, penalty functions are introduced to penalize solutions violating the constraints. Penalty functions associate constraints with (an) objective function(s) to construct (a) new objective function(s), making the satisfaction of the constraints self-included when optimizing the new objective function(s) [5]. Take equation 1 as an example. $f(x)$ in equation 1 is the objective function, and $g_1(x), g_2(x), g_3(x)$ and $g_4(x)$ are the constraints (inequalities) associated with the objective function. The constrained problem $\min f(x)$ can be solved as an unconstrained minimization problem as follows:

$$\min f'(x) = f(x) + \phi_i \sum_{i=1}^{4} d(g_i(x)) \tag{5}$$

$\min f'(x)$ represents the penalized function of $f(x)$ in equation 1. For $i = 1,2,3,4$, $d(g_i(x))$ is the penalty function and ϕ_i is the penalty coefficient. Equation 5 can be further expanded as follows:

$$\begin{aligned} &\min f'(x) = \min f(x) + [\phi_1 \max(g_1(x),0)] + [\phi_2 \max(g_2(x),0)] + ... \\ &[\phi_3 \max(g_3(x),0)] + [\phi_4 \max(g_4(x),0)] \end{aligned} \quad (6$$

3. OPTIMIZATION METHOD IN FDE 1.0

This section introduces the optimizer in FDE 1.0 and the parameter settings for an optimization run. Though algorithmic details of the optimizer are not covered, the properties and the recommended conditions for use are provided. Readers who are interested in further details on the optimizers can refer to the following paper: [1].

3.1 Surrogate Model Assisted Differential Evolution Algorithm for Filter Optimization (SMEAFO)

Focusing on filter optimization problems, the SMEAFO method achieves comparable design quality with standard evolutionary algorithms (EAs). At the same time, the SMEAFO method offers significant speed improvement by completing the optimization in a reasonable timeframe. Experiments show a 20-30 times speed improvement. SMEAFO has been shown to be compatible for the optimization of most types of filters without needing a special understanding of their specific properties. The SMEAFO framework consists of Gaussian Process (GP) machine learning as the surrogate modelling method, Differential Evolution (DE) as the global search engine and Gaussian local search for elaborate search in a local region of optimal solutions [6 – 9]. The combined operations of these mathematical models within the SMEAFO framework naturally divide the SMEAFO processes into two connected phases: exploration and exploitation.

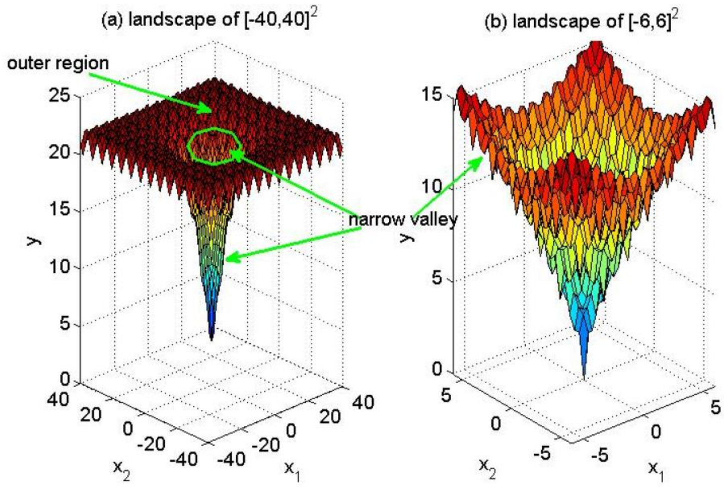

Fig. 3: Illustration of Filter Design Landscape Using the Ackley Function [10]

As illustrated in Fig. 3, filter optimization problems are characterized with a multiplicity of local optima and the optimal region is often located in a trough of the design landscape. With respect to filter optimization objectives, it can be said that the exploration phase of the SMEAFO seeks to find a near optimal region within the filter design landscape using a surrogate model-assisted global optimization technique. Once a near optimal region has been found, the exploitation phase is engaged to obtain the final optimal design from near-optimal designs using a surrogate model-assisted local search with decreased exploration ability. More details of the SMEAFO algorithm can be found in [1].

3.4 Parameter Settings for SMEAFO

Setting algorithmic parameters is often a challenge for filter designers unfamiliar with optimization methods. To address this, FDE 1.0 provides a highly interactive interface for user defined parameters. The performance of the SMEAFO algorithm available on the FDE 1.0 is related to the settings of the design variables and/or parameters. A detailed technical introduction to the parameter settings for the SMEAFO on the FDE 1.0 is provided as follows:

- ***Parameter Ranges***

  In some existing tools, the search range is set to, for example, 30% of the initial value. However, there is no substance to the notion that a larger value should induce a larger search range. Rather, the search range is decided by the quality of the initial design, and is not related to the absolute value of a design variable. Fig. 4 illustrates the design variable range setting of FDE 1.0.

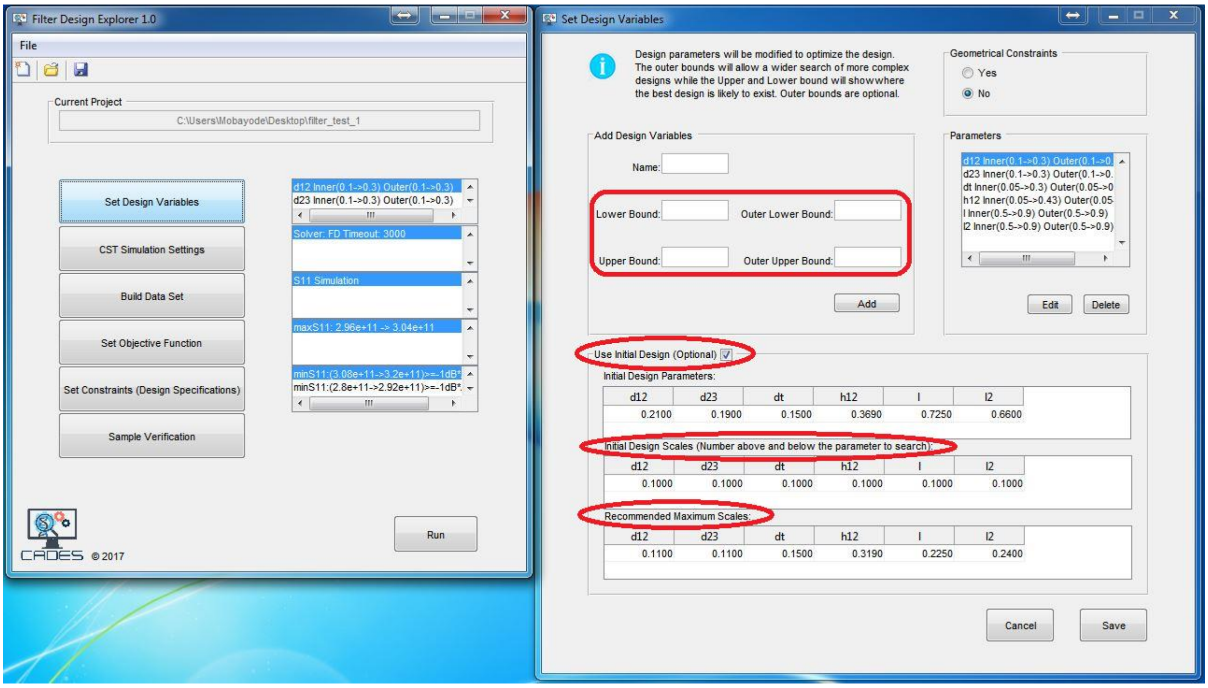

Fig. 4: FDE 1.0 Window for SMEAFO Parameter Settings

Recall that the optimum design of a filter is always in a very narrow valley (see Fig. 3) within the landscape of the design. At the risk of repetition, this is a considerable challenge for any search algorithm. Very often, designers are confused about setting the search range. The choice of a small range may cause some variables of the optimal design to be out of the search range making the optimization unsuccessful. On the other hand, the choice of a large range can make the valley even narrower causing the optimization to fail once more. To address this problem, FDE 1.0 allows the user to adopt two search ranges: inner and outer. The inner range is the range envisaged to have the optimum design (some variables may not but most are likely) while the outer range is the range outside of which, the optimum design is not possible. As such, the inner range should be much smaller than the outer range. The inner range is adopted for initialization. During the search, if newly generated candidate designs are characterized with variables outside the inner range, they will NOT be bounded to the inner range owing to the outer range specification. Only candidate solutions with some variables outside the outer range will be bounded to the outer range. In this way, the designer will find it easier to use their experience to find a good initial range and does not need to be worried about some optimal design variables not being in the initial range.

- ***Scale***

  'Scale' is the only critical parameter of FDE 1.0. To address the intricacies associated with the location of optimum designs within filter design landscapes, FDE 1.0 supports the specification of an initial design as shown in Fig. 4. For instances where an initial design is not readily available, FDE 1.0 is built to perform Latin Hypercube Sampling (LHS) to initially sample the design space to generate one. However, sometimes the resulting initial design from the LHS may be far away from the narrow valley where the optimal design lies. This may cause the consequent search to fail. From investigations and tests, it has been found that about 30% of the filter design exploration problems can be solved without an initial design. ***Therefore, even if the initial design is poor, it is strongly recommended for adoption.*** If an initial design is specified, the parameter 'scale' needs to be set. To sample the space, FDE 1.0 adds a determined Gaussian distributed random number to the initial design (x_0), $x_0 + N(0, scale)$, to perform the initial sampling. Thus, the essential information included in the initial design can be used. Experiments with some very difficult problems have revealed a success rate of more than 90%.

  It is intuitive that the worse the initial design, the larger the value of the scale. This is correct, and the following rules of thumb are provided for the user to decide the value of the scale:

  - In most cases, the same scale should be applied to all design variables. This is the default setting, but the user can change it freely.
  - A maximum scale is provided and it is suggested that the scale should not be larger than the maximum scale value. The reason is that when using the maximum scale value, more than 40% of the samples will be restricted by the inner range and are set to the inner range values. This is of course not good for initial sampling.
  - For difficult problems, the user can try several scales and run FDE 1.0 several times. If the optimization fails, the user can then try a smaller scale value and rerun. It has been found that many very difficult problems can be solved within three runs.

- **Penalty Coefficients**

  For constrained optimization problems, FDE 1.0 allows for two types of penalty coefficients: automatic and manual. The choice of penalty coefficients must be specified as shown in the FDE 1.0 window of Fig. 5:

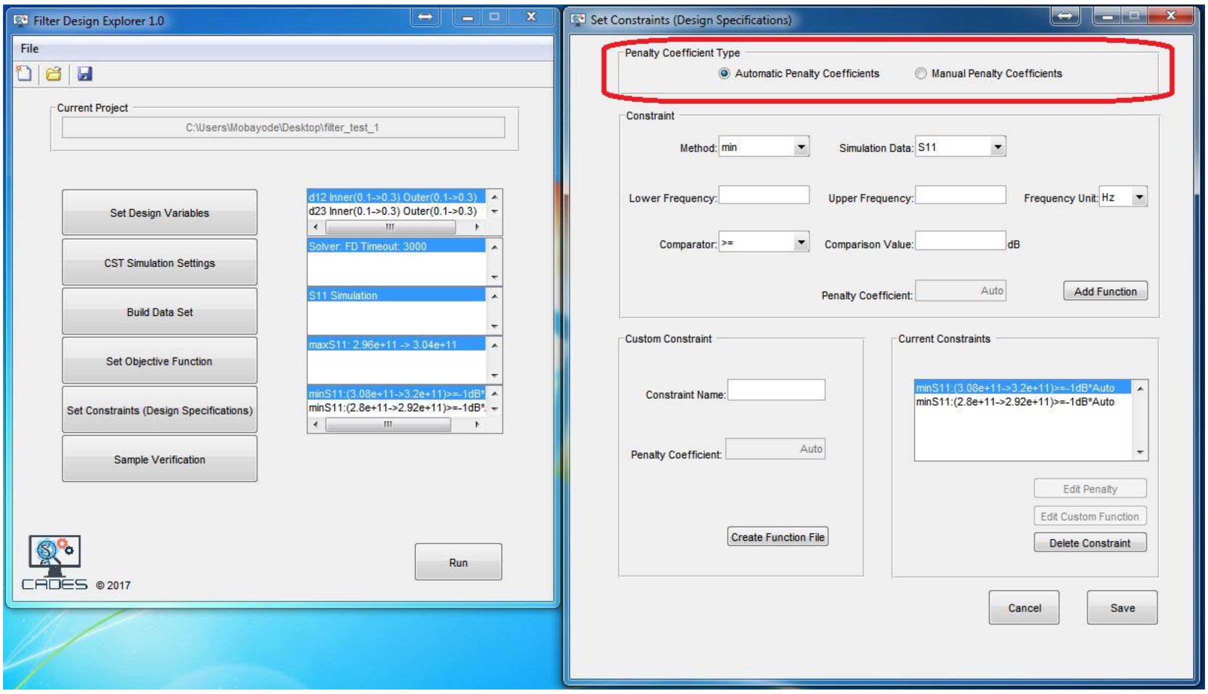

Fig. 5: ADE 1.0 Window for Penalty Coefficients' Settings

From Fig. 5, it can be seen that FDE 1.0 allows the user to automate settings for penalty coefficients to be adopted in the constraint function such that they augment the objective function. However, for manual input, it is strongly advised that the penalty coefficients be large enough to clearly discriminate the feasible and infeasible solutions.

4. BASIC USE OF FDE 1.0

This section provides the basic instructions on how to use FDE 1.0. FDE 1.0 is built in MATLAB. Hence, custom MATLAB functions can be used easily with the problems set in FDE 1.0. ***It is strongly recommended that the user watches the tutorial videos and employ this instruction set as a COMPLEMENTARY TOOL ONLY.***

4.1 Using the Software

FDE 1.0 offers a dynamic and easy-to-use GUI. First-time users are strongly encouraged to watch the tutorial videos.

- ***Run the Software***

  Before executing FDE 1.0, the user needs to check and confirm the data format for CST outputs or results. ***To avoid computational errors due to missing CST outputs or results, the storage mode for CST outputs or results must be set to ASCII and SQL.*** As shown in Fig. 6, this can be achieved by changing the general settings in the file options of CST Microwave Studio.

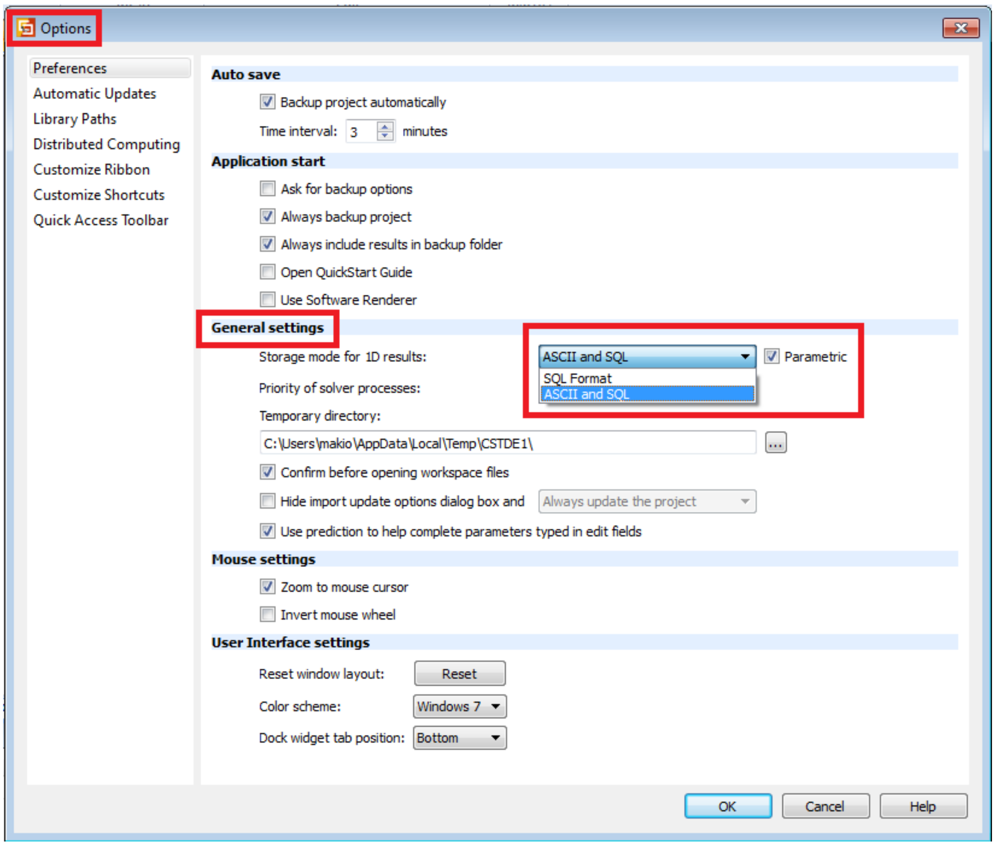

Fig. 6: CST Output Settings for FDE 1.0

- FDE 1.0 operates as a stand-alone application using a compatible version of the MATLAB runtime environment (https://uk.mathworks.com/products/compiler/mcr.html).
- To launch the software, simply double click on the FDE CADES icon after a successful installation (see appendix 1.0)

- ***Creating a New Project or Opening an Existing Project***

  - A new project is created and named, or an existing project opened, by clicking the appropriate icon, as shown in Fig. 7.

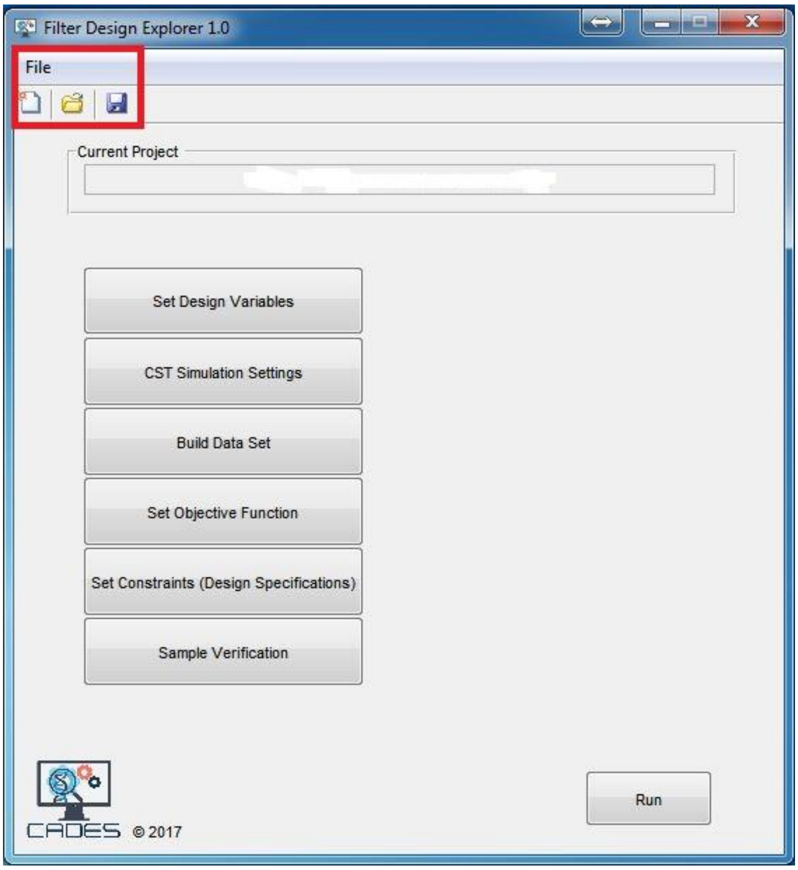

Fig. 7: Creating or Opening a Project File in FDE 1.0

- ***Setting the Design Variables***

  - To set the design variables, the lower and upper bounds are specified accordingly, as shown in Fig. 8.
  - ***For optimization problems involving CST models, it is strongly recommended that the user double-checks the parameter list of the CST model to ensure consistency with the variable names being specified in FDE 1.0.***
  - *(For fuller details on setting the design or parameters, see section 3.0.)*
  - *(See the video tutorials for more details "FDE 1.0".)*

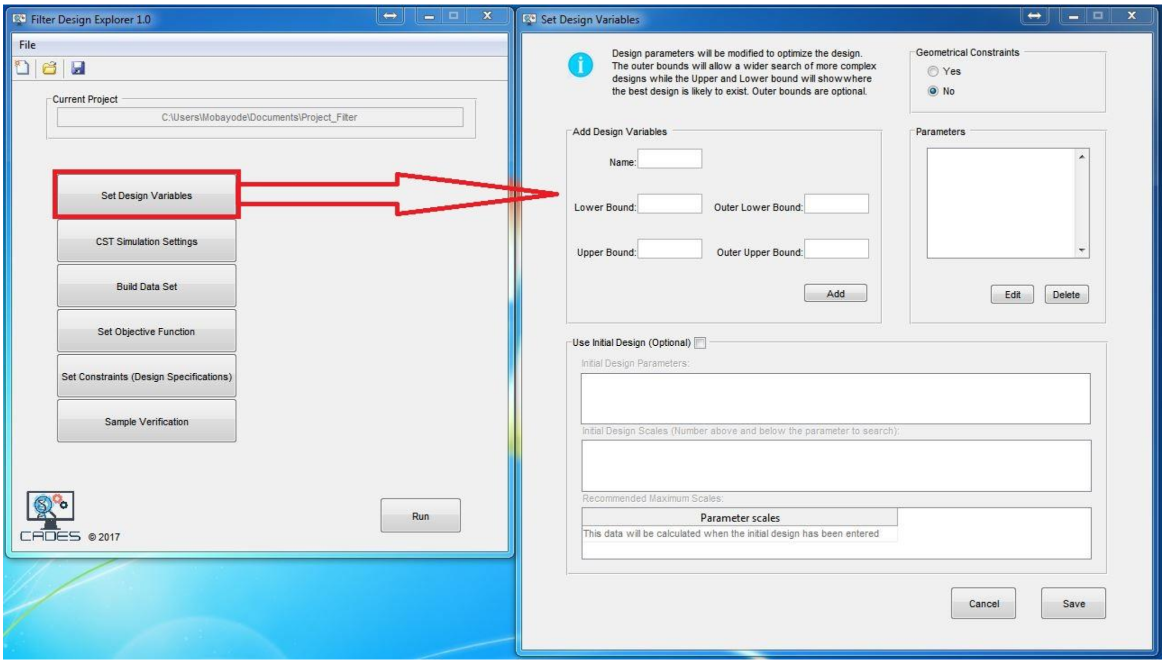

Fig. 8: Setting the Design Variables in FDE 1.0

- ***CST Simulation Settings***

    - As shown in Fig. 9, CST simulation settings can be set by selecting and specifying the installation path (CST Design Environment), the solver (Time Domain or Frequency Domain) and the CST Timeout. ***(The CST Timeout is recommended to be 2 - 3 times the approximate duration in seconds of the simulation time for the filter model in CST Microwave Studio.)***
    - Note that, if the wrong timeout duration is set, the CST Microwave Studio simulation will timeout during the FDE 1.0 runtime, hampering the optimization process. This will be repeatedly reported in an onscreen error message.
    - *(See the video tutorials for more details "FDE 1.0".)*

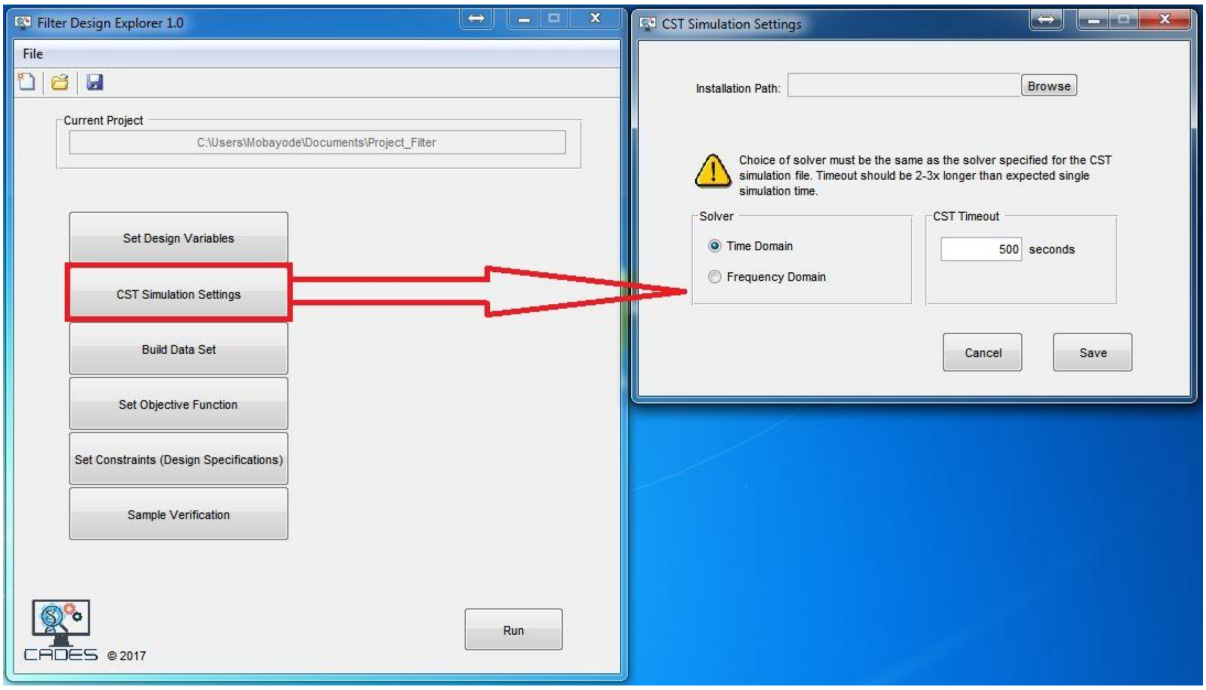

Fig. 9: CST Simulation Settings for FDE 1.0

- ***Build Data Set***

  - The data set is built by specifying the response name and selecting the CST model for the filter to be optimized *(refer to table 3 for the interpretation of CST responses)*.
  - As depicted in Fig. 10, a CST simulation is initiated seamlessly from FDE 1.0 to generate response signals unique to the selected CST model.
  - *(See the video tutorials for more details "FDE 1.0".)*

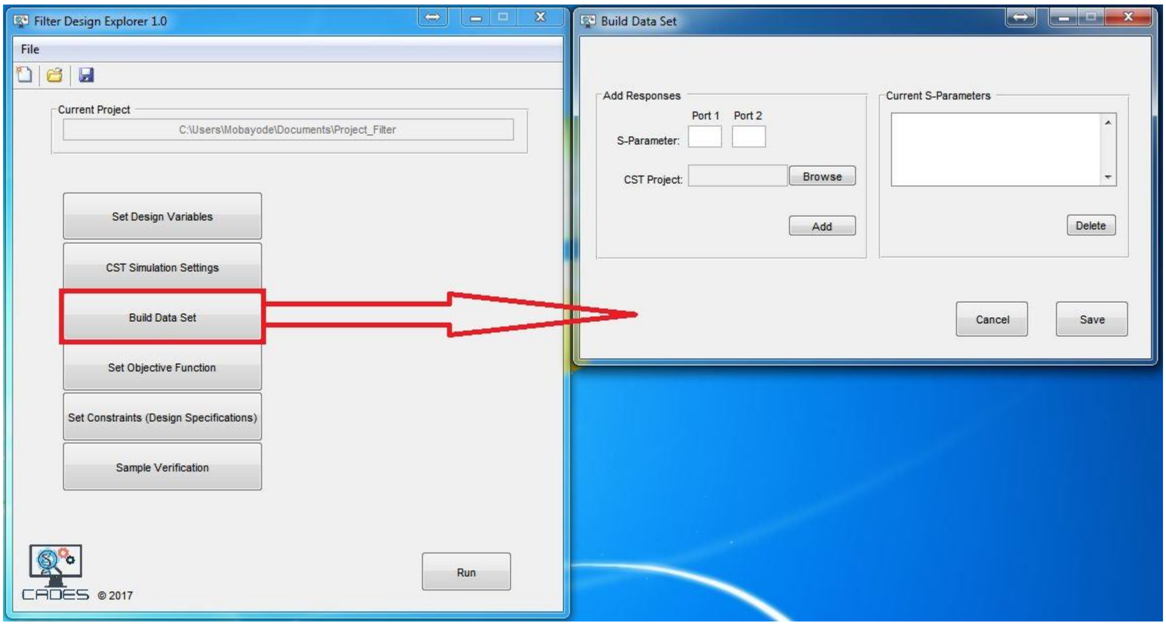

Fig. 10: Building the Data Set for FDE 1.0

- ***Set Objective Function(s)***
  - A shown in Fig. 11 and Fig. 12, users need only to specify the specific response signal from the simulated data and the frequency range for the optimization objective to set an objective function such as “minimize (max|S_{11}|)” from 296GHz – 304GHz.
  - However, FDE 1.0 also allows the flexibility of setting custom objective functions as shown in Fig. 11.
  - *(See the video tutorials for more details “FDE 1.0”.)*

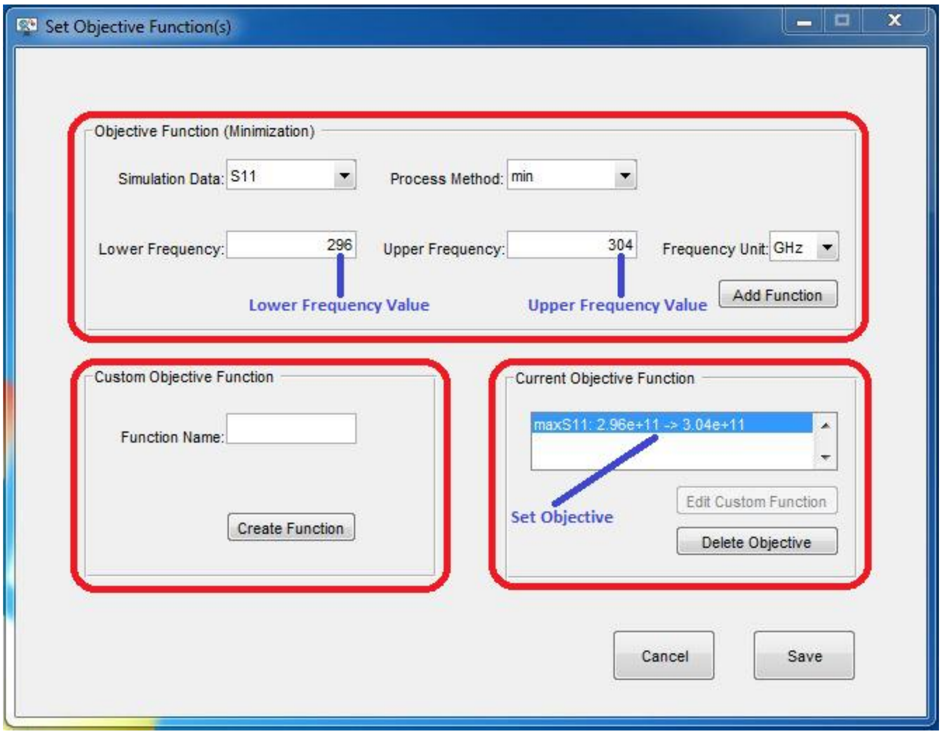

Fig. 11: Setting the Objective Function(s) for FDE 1.0 (User’s Specifications)

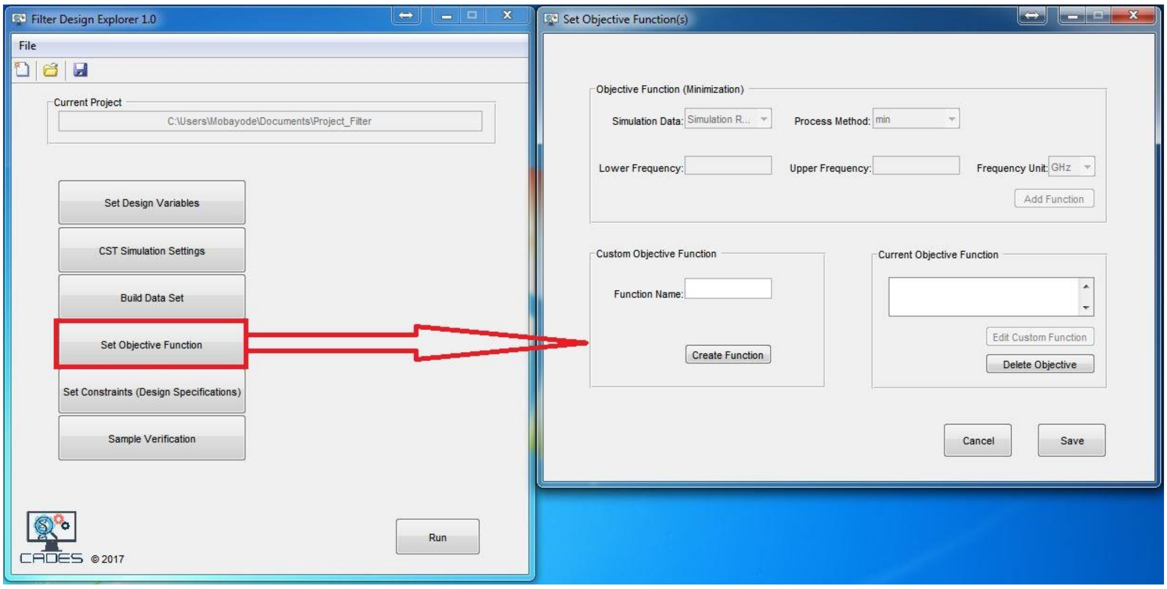

Fig. 12: Setting the Objective Function(s) for FDE 1.0

- ***Set Constraints (Design Specifications)***
  - Any constraints, such as min$|S_{11}|$ ≥ -1dB (at 280GHz – 292GHz and 308GHz – 320GHz respectively) shown in Fig. 13, can be defined and specified as required. The constraint name and associated penalty coefficient are to be specified as well. From Fig. 13 and Fig. 14, the following selections must be made and specified for each of the constraint functions built using the FDE GUI: method (minimum or maximum or mean or median); comparator (≤ or ≥); lower and upper bounds of frequency range; comparison value (in 'dB'); and a penalty coefficient.
  - The user can also adopt auto penalty coefficients and upload custom functions to suit the constrained optimization problem as shown in Fig. 13.
  - *(See sections 2.0 for a brief overview of penalty coefficients for constrained optimization)*
  - *(See the video tutorials for more details "FDE 1.0".)*

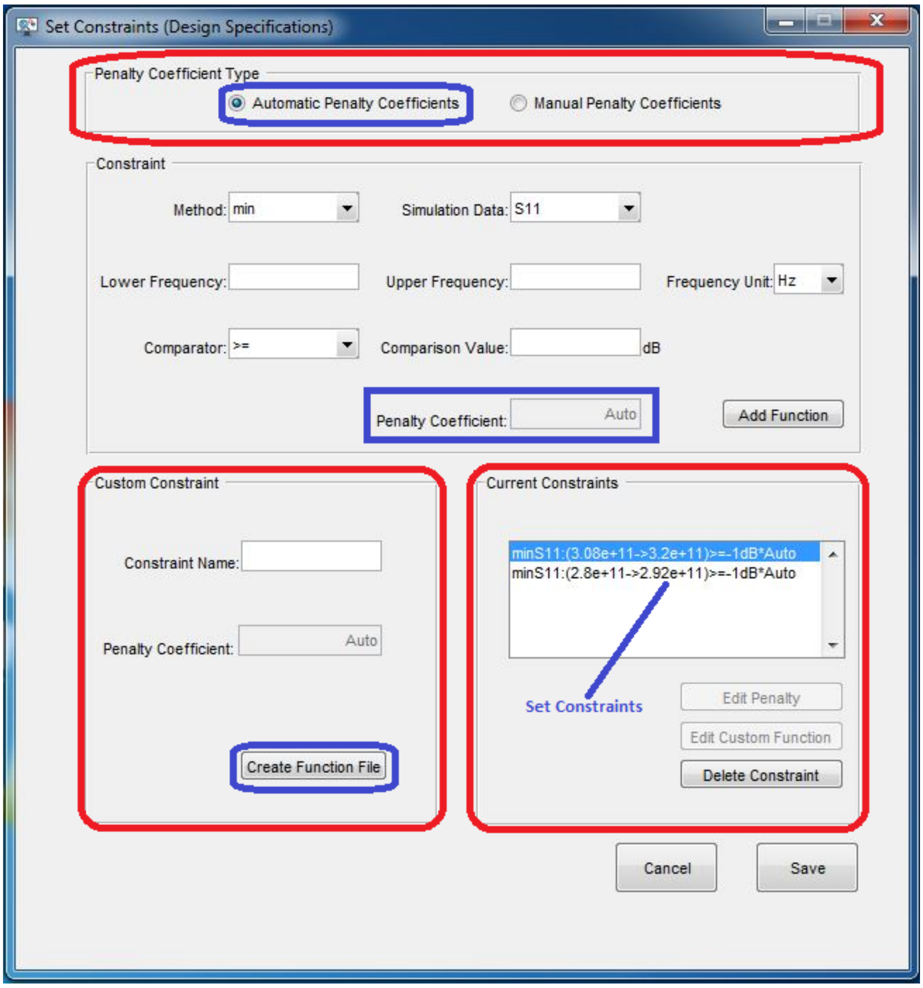

Fig. 13: Setting the Design Constraints in FDE 1.0 (User Specifications)

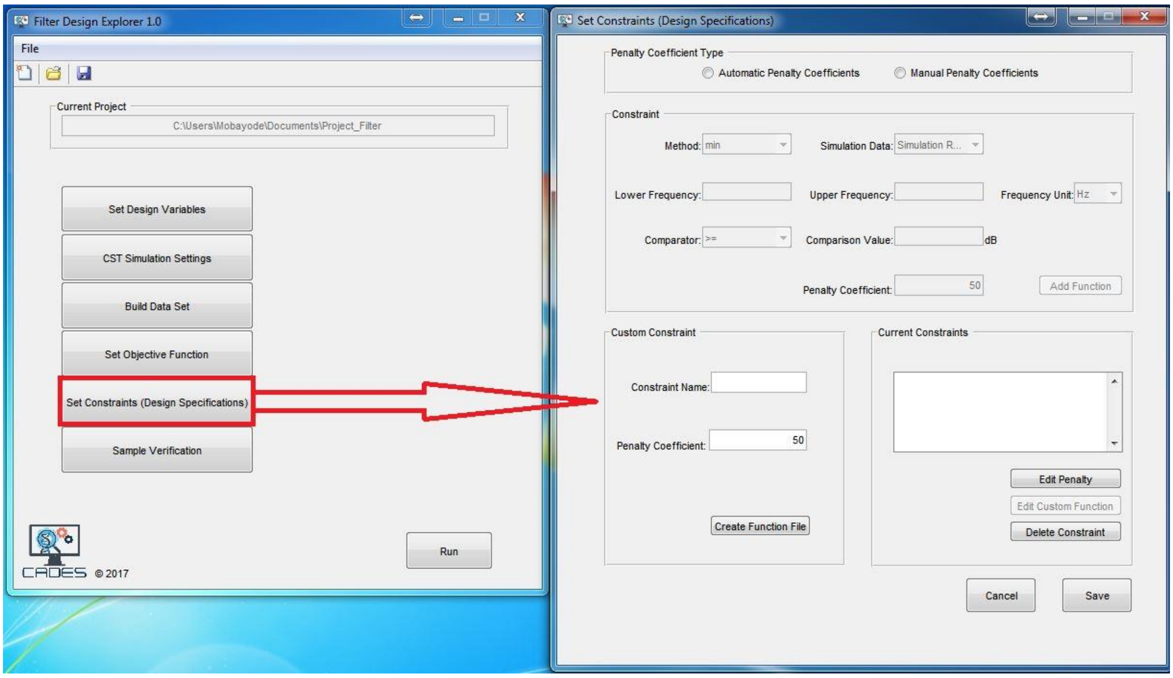

Fig. 14: Setting the Design Constraints in FDE 1.0

- ***Sample Verification***

  - Sample verification is carried out using a random set of variables within the specified bounds (lower and upper) of the design variables, either by entering the variables manually or through a "Batch Test", as shown in Fig. 15.
  - ***Note that, for the initial sample verification, the simulation process is carried out twice, to confirm that the defined inputs match the output. Upon confirmation, the simulation process runs only once for the subsequent sample verification.***
  - *(See the video tutorials for more details "FDE 1.0".)*

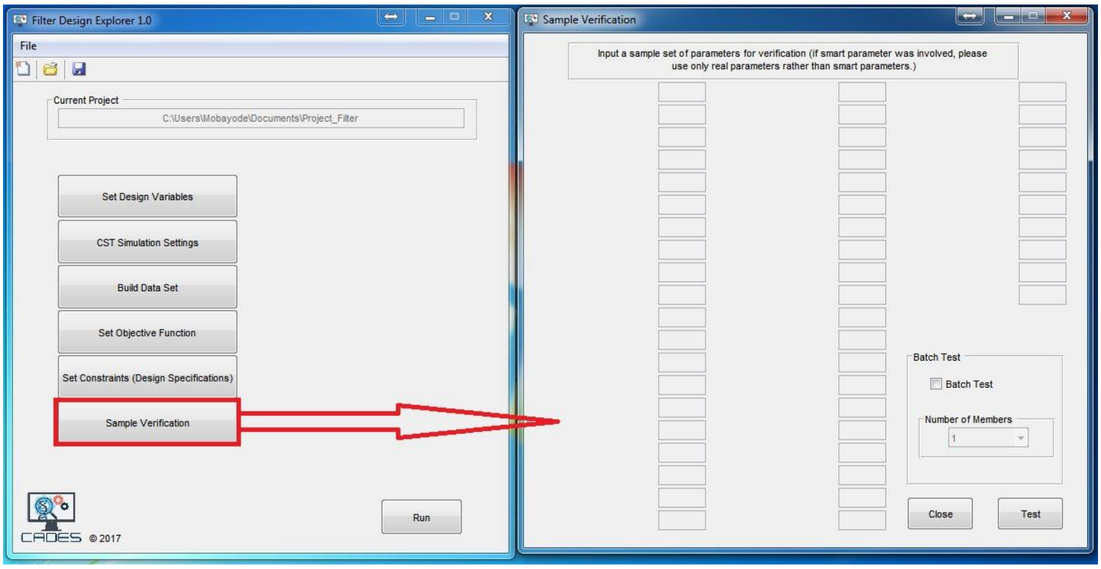

Fig. 15: Sample Verification in FDE 1.0

- ***Running the Optimization***

  - Once the sample verification tests succeed, optimization runs can be initiated or continued by clicking on the "Run" button as shown in Fig. 16. *(See sections 1.0 and 3.0 for fuller details of the optimization algorithm)*
  - *(See the video tutorials for more details "FDE 1.0".)*

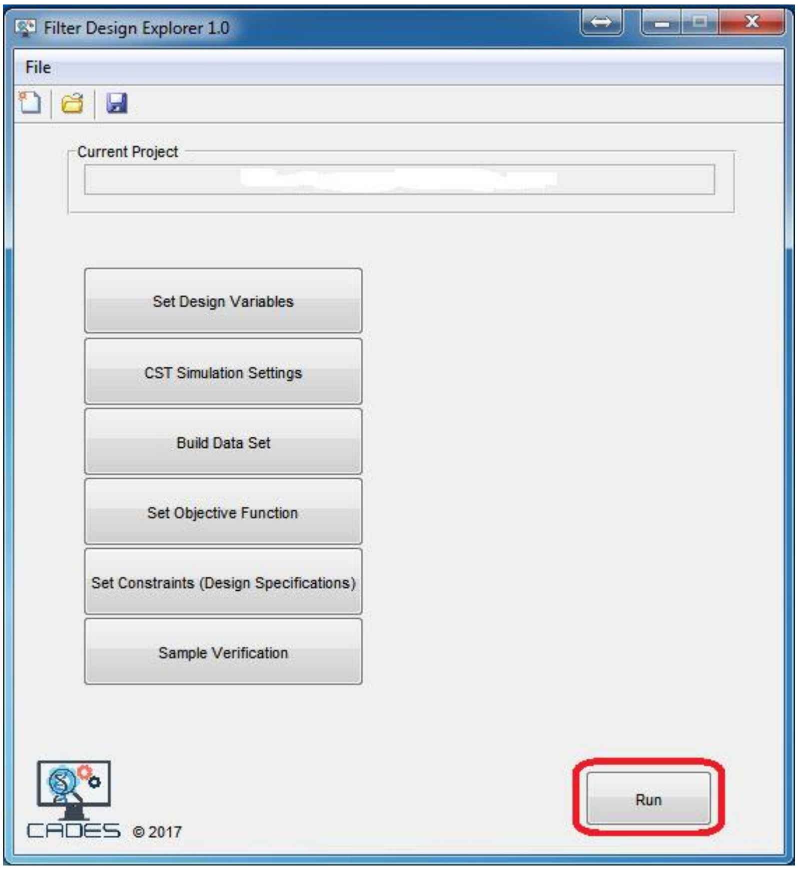

Fig. 16: Optimization Run in FDE 1.0

4.2 Essential Tutorials

- ***CST Response Signals: Interpretation and Translation***

  - Table 2 provides a list of response signals and their respective interpretations and translations, based on CST 2014-2016. *Please note that the table is to serve as a guide only.* ***It is strongly recommended that users always check to ensure the correct response signal output from CST is selected for the optimization process.***

Table 2: CST Response Signals (Interpretation and Translation)

RESPONSE SIGNALS	CST TO MATLAB (ADE 1.0) TRANSLATION	INTERPRETATION
Gain	**Gain (IEEE),Theta=x,Phi=y.sig**	Gain response signal, where 'x' and 'y' correspond to the specified angles of cut through the horizontal and vertical cut planes, respectively, of the radiation pattern
S – Parameter (Magnitude)	**ai(1)j(1).sig**	Magnitude form of the S-parameter (scattering parameter) response signal, where 'i' and 'j' correspond to the output and input ports, respectively.
S – Parameter (Complex)	**cSi(1)j(1).sig**	Complex form of the S-parameter (scattering parameter) response signal, where 'i' and 'j' correspond to the output and input ports, respectively.
S – Parameter (dB)	**di(1)j(1).sig**	'dB' form of the S-parameter (scattering parameter) response signal, where 'i' and 'j' correspond to the output and input ports, respectively.
S – Parameter (Phase)	**pi(1)j(1).sig**	Phasor form of the S-parameter (scattering parameter) response signal, where 'i' and 'j' correspond to the output and input ports, respectively.
Radiation Efficiency	**signal_default.sig**	Radiation efficiency response signal.
Total Efficiency	**signal_default_lf.sig**	Total radiation efficiency response signal.

- ***Observation and Interpretation of Results***

    - When viewing and estimating results from the optimization process, it is worth noting that two important factors, showing the progress of the optimization process, are convergence trend and diversity of the population.

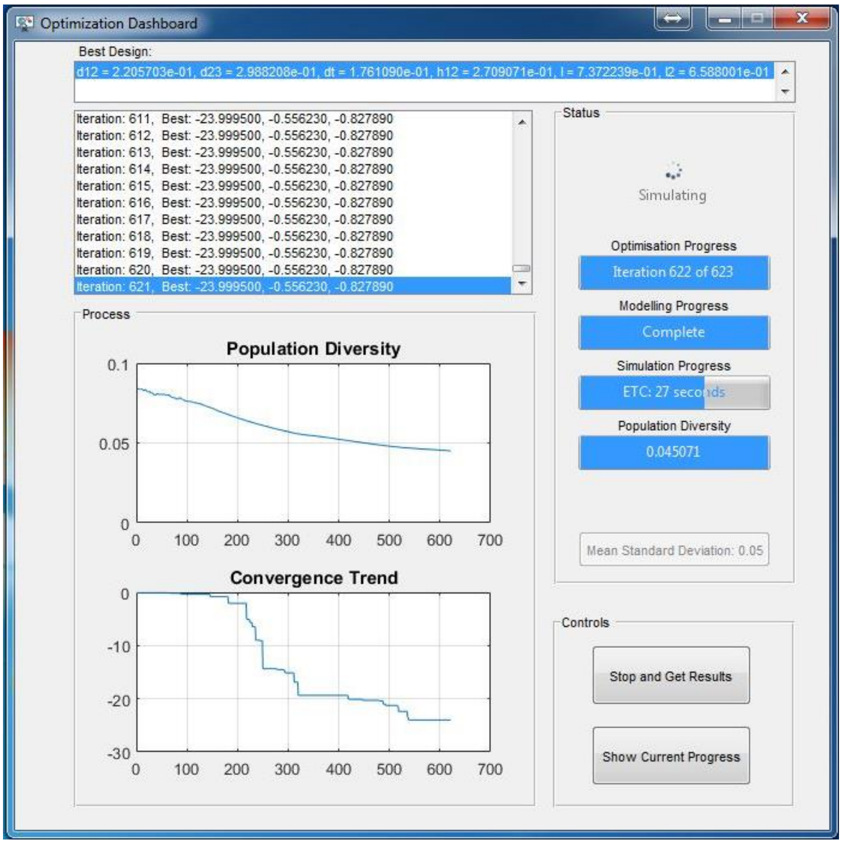

Fig. 17: FDE 1.0 Field Monitors for Optimization Process

- ***Population Diversity:*** Within the population, if the mean standard deviation among individuals is high, then the diversity is high. In a similar fashion, if the mean standard deviation among individuals of the population is small, then the diversity is low. For the test case shown in Fig. 17, the population diversity monitor indicates that there may be some room for further improvement.

- **Convergence Trend:** In FDE 1.0., the performance of the current best solution to the optimization problem can be easily observed on the convergence trend field monitor. For the test case given in Fig. 17, it can be seen that the result is already very good after 500 EM simulations. Hence, there is no need to continue, although we can expect some improvement until the population diversity falls very low.

- ***Optimization Timeout***

    - The FDE 1.0 optimization process is halted after the maximum number of iterations has been reached. If the field monitors (population diversity and convergence trend) strongly suggest that there is room for improvement, the optimization process can be continued by carrying out the following steps:

- ***Re-opening your Project File in FDE 1.0.:***

- To do this, click "Open" and select the "project.mat" file corresponding to the name of your current project in the appropriate folder: all settings will be loaded. By clicking "Optimize" in FDE 1.0, a new window will pop up, asking whether to continue an existing run or start a new run, as shown in Fig. 18. Select and click on "Yes" as shown in Fig. 16. ***(Ignore this step if your "project.mat" is already opened.)***

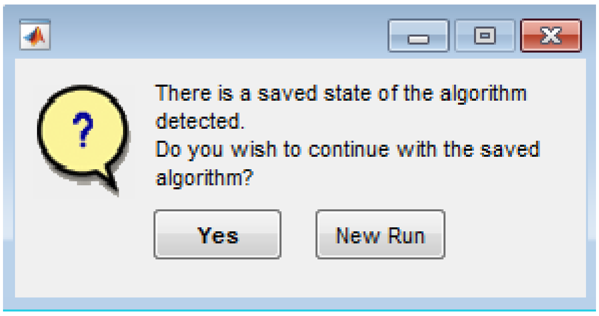

Fig. 18: FDE 1.0 Dialogue Box for Continuing an Optimization Run

- On clicking "Yes", a new window appears. Select and click on "Continue Optimization", as shown in Fig. 19.

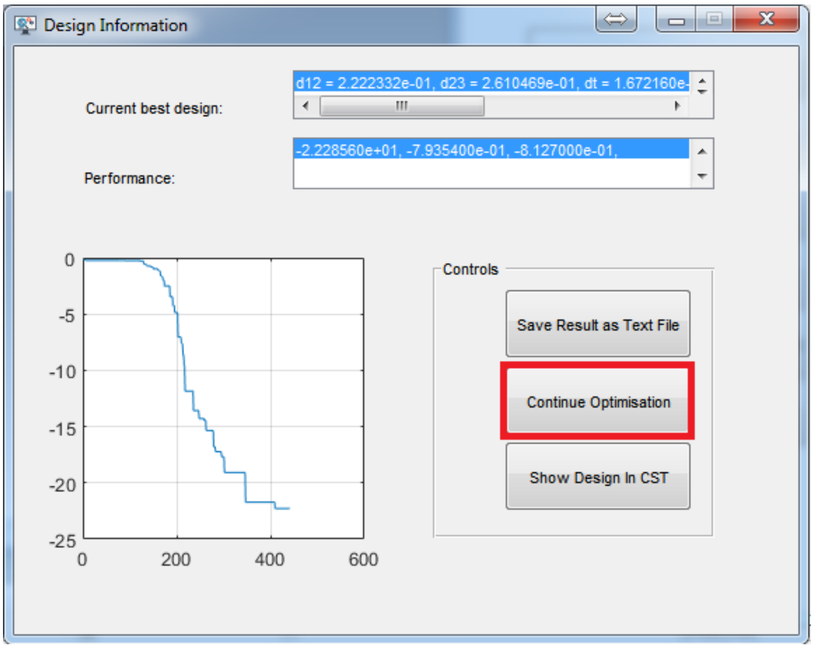

Fig. 19: FDE 1.0 Dialogue Box for Continuing Optimization Run (2)

- On clicking “Continue Optimization”, a new window appears. Specify the number of evaluations or iterations you require FDE 1.0 to continue running with and click “OK” to continue the optimization process, as shown in Fig. 20.

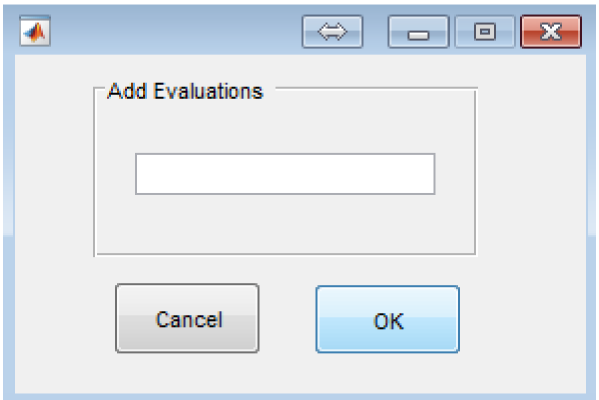

Fig. 20: FDE 1.0 Dialogue Box for Continuing an Optimization Run (3)

5.0 PROBLEM EXAMPLES

This section introduces some of the filter design optimization problems, which FDE 1.0 has been used to address. The problem examples presented here are for the following: WR-3 Band Waveguide Filter and Diplexer. It is strongly recommended to visit the "*FDE 1.0*" tutorial videos for a full analysis and description of these examples.

5.1 WR-3 Band Waveguide Filter

In this example, the design parameters of a WR-3 band waveguide filter have been adjusted to meet the following design specifications: the passband is from 296GHz to 304GHz (8GHz passband centered at 300GHz) and the $\max\left|(S_{11})\right|$ within the passband should be minimized; the stopbands are from 280GHz to 292GHz and from 308GHz to 320GHz where the $\min\left|(S_{11})\right|$ is to be better than -1dB. The optimization problem can be set as follows:

$$\begin{aligned} &\min imize \quad \max\left(\left|S_{11}\right|\right), \quad 296GHz - 304GHz \\ &s.t. \quad \min\left(\left|S_{11}\right|\right) \geq -1dB, \quad 280GHz - 292GHz \\ &\qquad \min\left(\left|S_{11}\right|\right) \geq -1dB, \quad 308GHz - 320GHz \end{aligned}$$

Fig. 21 shows the 3-D geometry the WR-3 band waveguide filter. The range of design variables adopted for the optimization of the WR-3 band waveguide filter can be found in Table 3:

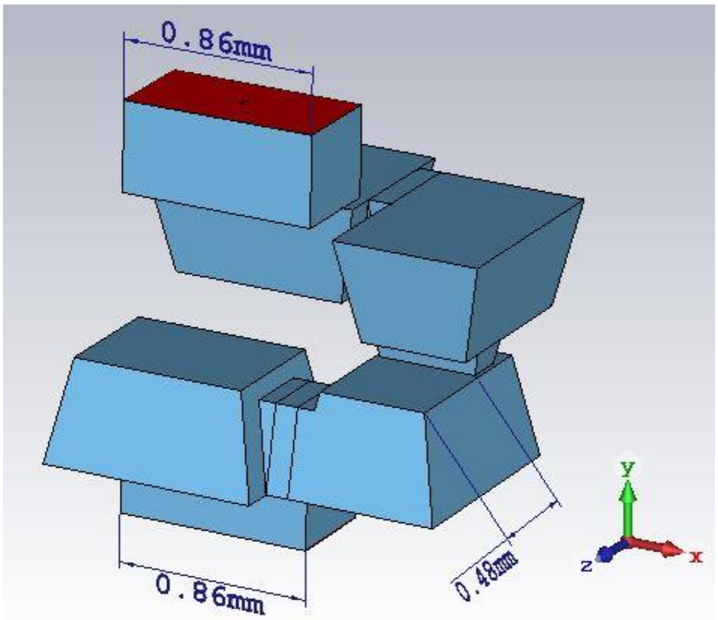

Fig. 21: 3-D Geometry of the WR-3 Band Waveguide Filter [1]

The range of design variables adopted for the optimization of the DRA can be found in Table 3:

Table 3: Design Variables for Design Exploration of the WR-3 Band Waveguide Filter

Variables	d12	d23	dt	h12	l	l2
Lower Bound (mm)	0.1	0.1	0.05	0.05	0.5	0.5
Upper Bound (mm)	0.3	0.3	0.3	0.3	0.9	0.9
Initial Design (mm)	0.21	0.19	0.15	0.369	0.725	0.66
Recommended Maximum Scales	0.11	0.11	0.15	0.319	0.225	0.24
Adopted Scales	0.1	0.1	0.1	0.1	0.1	0.1

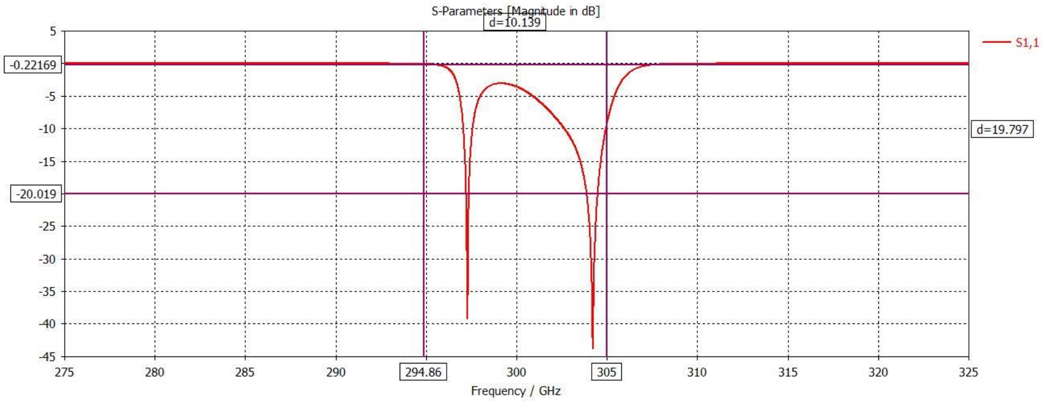

Fig. 22: Original S-Parameter Plot for the WR-3 Band Waveguide Filter

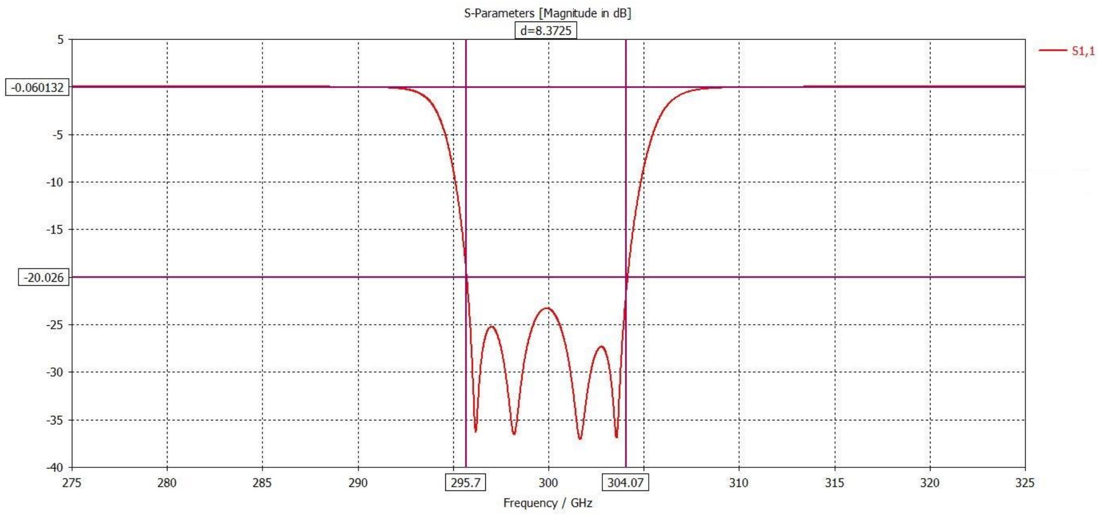

Fig. 23: Optimized S-Parameter Plot for the WR-3 Band Waveguide Filter

Fig. 22 and 23 show the CST simulation results before and after optimization of the WR-3 band waveguide filter problem respectively using FDE 1.0. From table 3, it can be seen that an available initial design and a scale of 0.1 was adopted for the optimization after scale values of 0.13, 0.1 and 0.08 had been tried. In Fig. 23, the optimum reflection coefficient values between 296GHz and 300GHz over a bandwidth of 8GHz at a center frequency of 300GHz can be found to be less than -20.0dB with four resonances at 296GHz, 298GHz, 301GHz and 303GHz having approximate reflection coefficient values of less than -36dB. Also from Fig. 23, the stopbands are clearly visible in the specified frequency regions: 280GHz to 292GHz and 308Hhz to 320GHz. These values validate realization of the optimization objective strongly. A full description, and step by step illustration, of the solution to the WR-3 band waveguide filter optimization problem using FDE 1.0 has been made available by the CADES research center. This can be found using this link: CADES Research Center (2017), Waveguide Filter Optimization Using FDE 1.0

REFERENCES

[1]. B. Liu, H. Yang and M. Lancaster (2017), "Global Optimization of Microwave Filters Based on a Surrogate Model Assisted Evolutionary Algorithm", *IEEE Transactions on Microwave Engineering*, (in press).

[2]. B. Liu, G. G. E. Gielen and F. V. Fernandez (2014), "Basic Concepts and Background". In Automated Design of Analog and High-frequency Circuits: A Computational Intelligence Approach. Berlin: Springer-Verlag Berlin Heidelberg, pp. 1 – 16.

[3]. A. Hedar, (2005, Nov.), *Test Problems for Constrained Global Optimization: G12 Problem.* [Online] Available at:http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page4221.htm

[4]. A. Ibrahim, S. Rahnamayan, M. V. Martin and K. Deb (2016), "3D-RadVis: Visualization of Pareto Front in Many-Objective Optimization", *IEEE World Congress on Computational Intelligence (IEEE WCCI) Proceedings*, pp. 736-745.

[5]. A. E. Smith and D. W. Coit (1996), "Penalty Functions". In Handbook of Evolutionary Computation. T. Baeck, D. Fogel and Z. Michalewicz Ed. Bristol: Institute of Physics Publishing and Oxford University Press, section C 5.2.

[6]. B. Liu, S. Koziel and N. Ali (2016), SADEA-II: A Generalized Method for Efficient Global Optimization of Antenna Design, *Journal of Computational Design and Engineering*, Elsevier, (In press). Available: http://www.sciencedirect.com/science/article/pii/S2288430016300859

[7]. N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan and P. K. Tucker (2005). Surrogate-based Analysis and Optimization (205), *Progress in Aerospace Sciences*. vol. 41, no. 2005, pp. 1 – 28.

[8]. R. Storn and K. Price (1997), Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, Springer, vol. 11, no. 4, pp. 341 – 3599.

[9]. X. Yao, Y. Liu and G. Lin (1999), Evolutionary Programming made faster. *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82 – 102.

[10]. M. Jamil and X. S. Yang (2013), A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150 – 194.

APPENDIX 1.0 : FDE 1.0 INSTALLATION QUICK GUIDE

- **To install the software, you need to download and run the setup file on your PC. The icon for the setup file is shown in Fig. 30:**

Fig. 30: FDE Setup Icon

- **As shown in Fig. 31, the 'End User License Agreement' must be accepted to continue the installation process:**

Fig. 31: FDE End User License Agreement (EULA)

- By default, the FDE is installed on the path of executable files on the PC. However, a different or unique location or path can be specified on the PC by using the “Browse” button. Fig. 32 can be referred to.

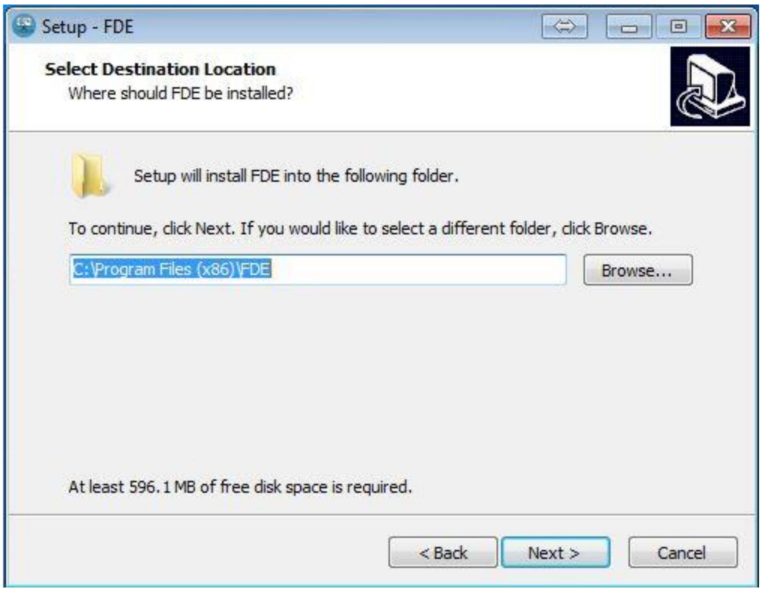

Fig. 32: FDE Installation Location / Path

- Once the installation location or path has been specified, the installation settings are confirmed by clicking the “Install” button. To change or edit installation settings, the “Back” button can be clicked. Fig. 33 can be referred to.

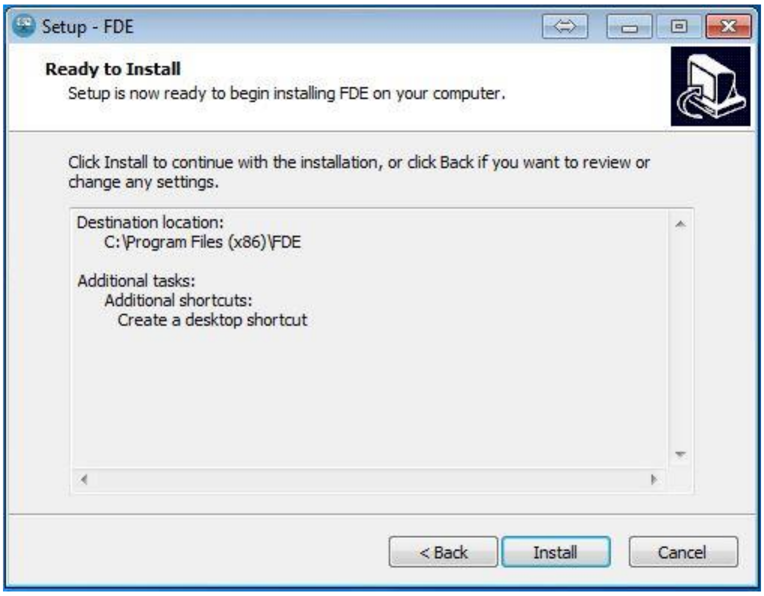

Fig. 33: FDE Installation Settings Confirmation

- After clicking the “Install” button, the FDE installation commences automatically as shown in Fig. 34.

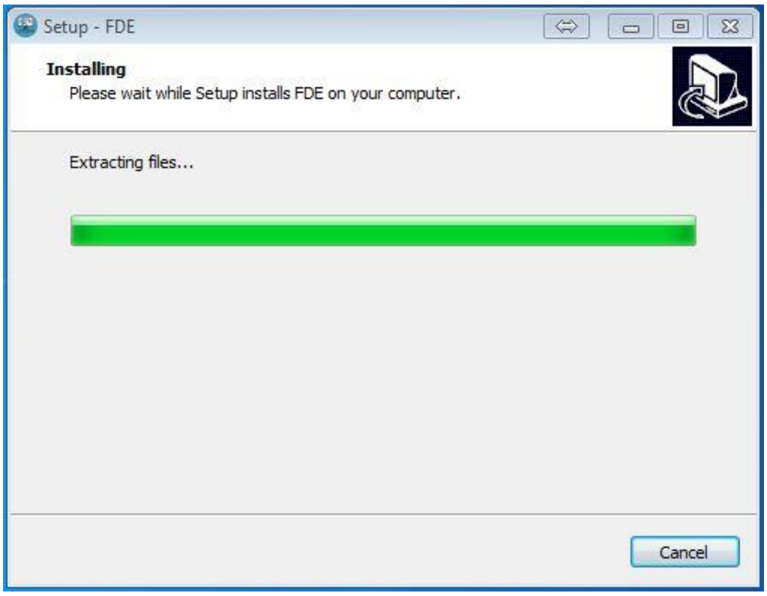

Fig. 34: FDE Installation Commencement

- As shown in Fig. 35, the run of the FDE’s setup file is packed to also install a compatible version of the “MATLAB Runtime” automatically.

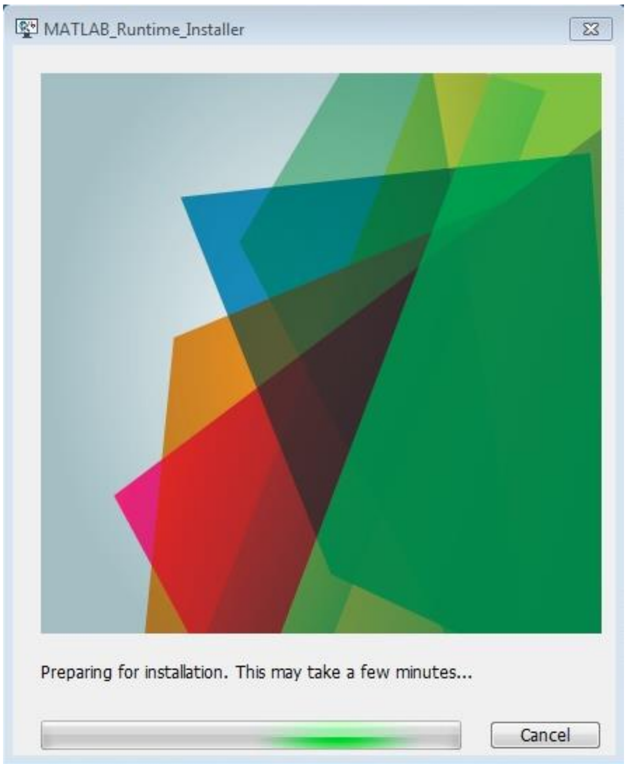

Fig. 35: ADE MATLAB Runtime Installation

- **Once a compatible version of the “MATLAB Runtime” has been installed, the installation process of the FDE is completed by clicking the “Finish” button as shown in Fig. 36.**

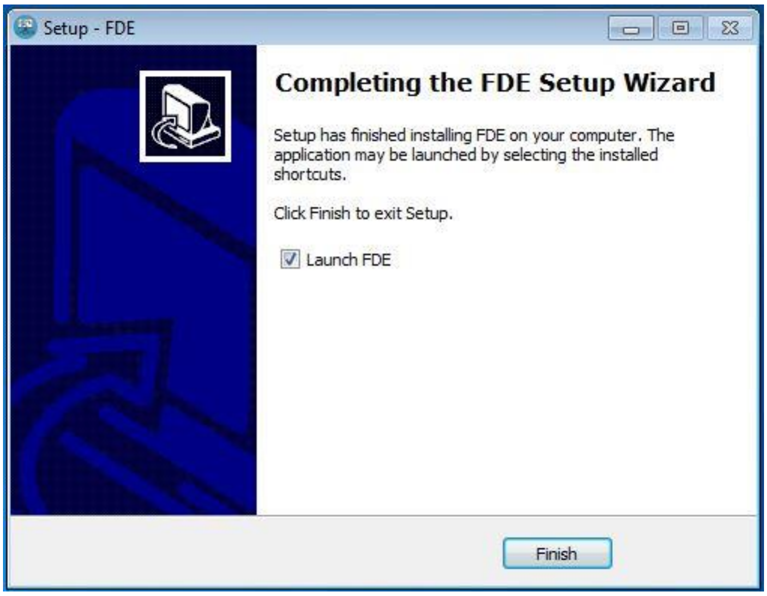

Fig. 36: FDE Installation Completion