“Damage Pursuit: User Manual”

1 Introduction

The goal of Nondestructive Defect Localization (NDL) is to detect the locations of anomalies in a structure without using the knowledge of material properties of the medium. A recently developed approach to this end consists of propagating acoustic waves through the structure and acquiring measurements of the propagating wavefield at discrete points over the surface of the structure by means of a scanning laser interferometer. The characteristics of the wavefield depend on the material properties of the structure which results in different wavefield characteristics at the atypical regions of the structure. The difference between the characteristics of the propagating wavefield at defect locations and the typical propagation pattern over the bulk of the structure will help extracting information about structural defects.

1.1 Damage Pursuit

Damage pursuit is a Matlab-based software for detecting and triangulating material anomalies in physical structures. It works by decomposing the spatiotemporal displacement measurements of the structure, which are resulted from a propagating wave, into a spatially-smooth component associated with the undamaged bulk of the medium, and a spatially-localized component caused by existing anomalies. In order to perform the decomposition, Damage Pursuit seeks efficient representations of the smooth and spatially-localized components in terms of appropriate bases (also called dictionaries), which are chosen based on the overall behavior of the respective component.

2 Matlab GUI

The developed GUI interface makes the use of the package more convenient. Here, we illustrate the steps of using the GUI for running the software.

1. Type `DamagePursuit_Gui` in the Matlab command line to make the GUI interface appear.
2. Press the `Load File` button to load the data. The data should be in the mat format.

   Damage Pursuit assumes the measurements are stored in the form of a data cube, with the first and second dimensions indicating the sizes of the acquisition grid defined over the surface of the structure (i.e. the size of images) and the third dimension denoting the number of taken snapshots. Essentially, the slices of the data cube correspond to the snapshots of the dynamic response at different time instants.
3. Enter the width and height of the test specimen in the boxes for `Lx` and `Ly`, respectively.
4. In the `Smooth Dictionary` panel, select the type of the basis for the spatially-smooth component of the response. The user can either select the discrete cosine transform (DCT) matrix or the discrete Fourier transform (DFT) matrix as the basis.
5. In the `Sparse Dictionary` panel, select the dictionary type for the spatially-localized component of the decomposition; the possible choices are the canonical (Dirac) basis and the two-dimensional Marr wavelet.

6. In the `Spt Rsl` box, the user can enter an integer d which will be employed to spatially partition the image domain (i.e. the structure surface) into squares of sizes $d \times d$. When the Dirac basis is selected as the dictionary for the smooth component, the software will seek anomalies in the form of $d \times d$ squares of pixels.

7. If the two-dimensional Marr Wavelet dictionary is selected as the sparse dictionary, the radius `sigma` of each wavelet function has to be specified in the corresponding box.

8. `tau1` and `tau2` are non-negative scalar parameters, which are technically called *regularization parameters*. They are required from the user and play a critical role in accurate localization of the anomalies.

   The regularization constants specify the trade-off between the goodness of the fitted decomposition model and the parsimony in representing the decomposition components in terms of their respective bases. Roughly speaking, by decreasing the value of these parameters, the model fitting error will be reduced but the representations will become more dense, i.e. more basis elements will be involved in representing the respective components.

9. The user can run the package for all the snapshots of the data by selecting the `All Frames` button or only execute it on a certain subset of frames. In the latter case, the numbers for the starting and ending frames should be specified in the boxes.

10. The user can look at the demixing plots for any of the processed frames.

Figure 1 shows the GUI window with plots of the constituent components resulting from the decomposition of an example image. Loading the data and setting configurations for the experiment should be done in the left panel of the GUI window; the right panel shows the plots resulting from the decomposition. The top subplot in the right panel shows the original wavefield snapshot whereas the two bottom subplots display the resulting components. The left and right subplots show the spatially-sparse and -smooth components, respectively. The signature of anomaly is clearly visible in the subplot for the spatially-sparse component.

3 Command Line Usage

Damage Pursuit assumes the measurement data is given as a data cube $\boldsymbol{Y}$ of size $d_1 \times d_2 \times T$ where d_1 and d_2 are image sizes, and T is the number of measurement snapshots. The algorithm is endowed with the possibility of shifting a time window, of length $T_w \ll T$, over the slabs of the data cube $\boldsymbol{Y}$ and selecting a subset $\boldsymbol{Y}_\tau \in \mathbb{R}^{d_1 \times d_2 \times T_w}$ at every step τ where $1 \leq \tau \leq \lfloor \frac{T}{T_w} \rfloor$. Beginning with $\tau = 1$, $\boldsymbol{Y}_\tau$ wil be taken as the observation matrix that is to be decomposed. After the decomposition is accomplished for $\boldsymbol{Y}_\tau$, the time window will be shifted by T_w slices and the next sub-tensor $\boldsymbol{Y}_{\tau+1}$ will be taken. The time windowing will continue until the last sub-tensor, which corresponds to $\tau = \lfloor \frac{T}{T_w} \rfloor$, is decomposed. The value of T_w can be given by the user; otherwise it will be set to one by default. Damage Pursuit assumes a partition of the spatial domain into $d_i \times d_i$ regions.

- **Example One:** By Typing

  ```
  Damage_Pursuit(Y,'sgm',1e-3,'Lx',1,'Ly',0.5,'D1_Type',0,'D2_Type',1,...
                 'tau1',1e-5,'tau2',2e-5);
  ```

  the package will take `Y` as the data cube. The names of the rest of the parameters should be specified in quotation marks. The unspecified parameters will be set to their default values according to Table 1. For instance `Spt_Rsl` and `Tmp_Rsl` will be set to one by default after executing the above command.

Table 1 lists the inputs and outputs of the software.

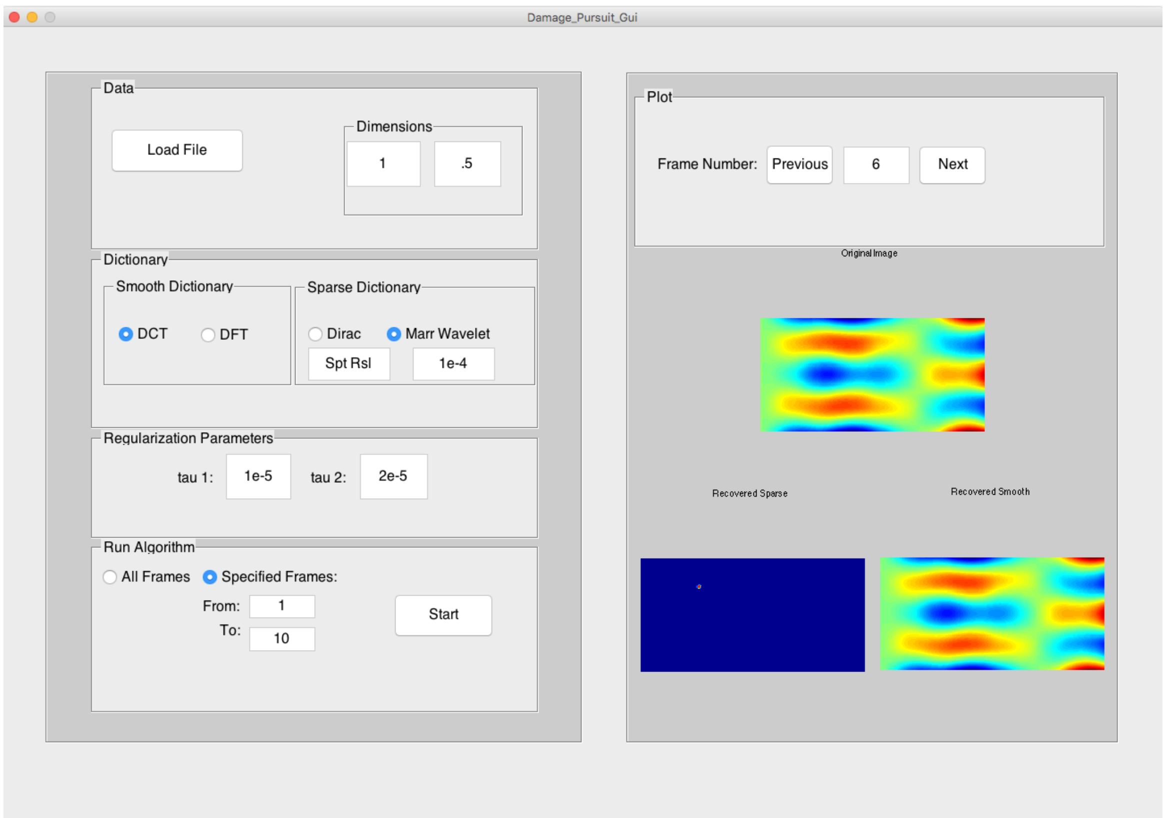

Figure 1: The GUI window with plots of the constituent components resulting from the decomposition of an example image. Loading the data and setting configurations for the experiment should be done in the left panel of the GUI window; the right panel shows the plots resulting from the decomposition. The top subplot in the right panel shows the original wavefield snapshot whereas the two bottom subplots display the resulting components. The left and right subplots show the spatially-sparse and -smooth components, respectively. The signature of anomaly is clearly visible in the subplot for the spatially-sparse component.

MATLAB Notation	Description	Default Setting
D1_Type	The dictionary for the spatially-smooth component is specified by D1_Type. By setting the parameter to 0 and 1, one can set the basis to the DCT or DFT, respectively. The option to use a different basis is available by setting this parameter to 2	The default value is 0, i.e. DCT is the default basis for the spatially-smooth component.
D1	The sparse representation basis of the spatially sparse component of the decomposition. When D2_Type = 2, then D2 should be given as a handle to a function which computes products of the form $\boldsymbol{D_2 x}_2$.	The Dirac dictionary is the default.
D1T	Inverse of the sparse representation basis. When D1_Type = 2, then D1 must be inputted as a handle to a function that computes products of the form $\boldsymbol{D}_1^T\boldsymbol{x}$.	The Inverse DCT is the default setting for $\boldsymbol{D}_1^T$.
D2_Type	The dictionary for the spatially-sparse component is specified by D2_Type. By setting the parameter to 0 or 1, one can set the dictionary to the Dirac or Marr Wavelet, respectively. Using a different dictionary is possible by setting this parameter to 2.	The default value is 0, which means Dirac basis is the default dictionary for the spatially-sparse component.
D2	The sparse representation basis of the spatially smooth component of the decomposition. When D1_Type = 2, then D1 should be given as a handle to a function which computes products of the form $\boldsymbol{D_1 x}_1$.	The Discrete Cosine Transforms (DCT) is the default. The Discrete Fourier Transform (DFT) is the other sparsifying transformation that is provided by the package.
D2T	Inverse of the sparse representation basis. When D2_Type = 2, then D2 must be inputted as a handle to a function that computes products of the form $\boldsymbol{D}_2^T\boldsymbol{x}$.	The Dirac dictionary is the default setting for $\boldsymbol{D}_2^T$.
Tmp_Rsl	Length of the Time Window T_w	Default value is one.
Spt_Rsl	Length d_i of the sides of every rectangular region in the partitioned image. Group sizes in vectorized images will be d_i^2.	Default value is one.
tau1	τ_1 is the regularization constant for the spatially-smooth component.	Default value is 0.005
tau2	τ_2 is the regularization constant for the spatially-sparse component.	Default value is 0.005
Margine	This number indicates how many partitions from each side of the image should be considered as boundary regions. Norm of such regions will be penalized more aggressively.	Default value is zero.
tau3	τ_3 is the regularization constant for the marginal regions.	Default value is $\tau_3 = 2\tau_2$.
Dws_fac	This number is used as the rate to uniformly down-sample images. Down-sampling might be used to lower the computational complexity of the algorithm.	Default value is one, i.e. no down-sampling is applied.

Table 1: Input Arguments

MATLAB Notation	Description	Default Setting
`sgm`	`sgm` is the variance of two-dimensional Marr wavelet functions. This parameter is only needed when Marr wavelet is used as the basis for the spatially-sparse component.	Default value is 10^{-5}.
`Lx`	`Lx` is the height of the structure. This parameter is only needed for making the Marr wavelet dictionary.	
`Ly`	`Ly` is the width of the structure. This parameter is only needed for making the Marr wavelet dictionary.	
`Initial`	The point at which the algorithm is initialized. It must be one of $\{0, 1, 2, 3\}$ • 0 : The problem of the first T_w time slices will be initialized at zero while the next ones will be warm started. • 1 : Zero initialization for all the frames. • 2 : The problem of the first T_w frames will be initialized randomly but the next ones will be warm started. • 3 : Random Initialization for all the frames. Both $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ will be initialized according to the selected rule.	Default value is zero.
`Max_Itr`	Maximum number of iterations that the algorithm takes.	Default value is 500.
`Trm_Crt`	Termination criterion to use: • 0 : Norm of the distance of two consecutive iterates relative to the norm of one iterate is used as the termination criterion. • 1 : The norms of the residuals of the gradient optimality condition are used as the stopping criterion.	Default value is zero.
`eps`	It determines the required accuracy at the solution of the algorithm.	Default value is 10^{-4}.
`Plot_flag`	If this flag is one, the reconstructed coefficients will be plotted.	Default value is zero.

Table 2: Input Arguments (Cont.)