

# SimpleDIVA User Manual

---

Authors: Frank H. Guenther, Alfonso Nieto-Castañón, Riccardo Falsini, Elaine Kearney, & Hasini R. Weerathunge

Version 2.0, Released March 3, 2023

## Contents

---

### SimpleDIVA User Manual

- Contents

- Quick Start Guide

- Introduction

- Installation

  - Windows

  - Mac

  - Linux

- Examples

  - Individual data or mean data across subjects

  - Data from multiple subjects across different experimental paradigms

- Advanced Options

  - Additional parameters

  - Regularization

  - Auditory target estimation

  - Baseline seeding

- Uninstall SimpleDIVA

  - Windows

  - Mac

  - Linux

- FAQ

  - Can I fit blocked data as well as individual trial data?

  - When would I want to fix the model parameters? And what values do I use?

  - Can I fit data from an experiment that used masking noise?

  - What about missing data?

  - Where can I find the latest version of the app?

  - How do I cite the software?

- References

- License

- Advanced Information for BU Speech Lab Members

  - Parallelizing simulations on the shared computer cluster (SCC)

# Quick Start Guide

---

- Step 1: [Install](#) the SimpleDIVA application.
- Step 2: **Format your dataset** as a csv file with one row per trial of the experiment (adaptive data) or one row per timepoint of the experiment (reflexive data). Column 1 contains the perturbation value for each trial/timepoint (as a decimal percentage; e.g., a 30% downward shift is -0.3). Each subsequent column contains response data. When modeling adaptive data, each column of data is from an individual subject. When modeling reflexive data, each column of data can be from an individual subject (averaged over multiple trials) or from individual trials.
- Step 3: Start the **SimpleDIVA** app.
- Step 4: Under '**Experiment Type**', select '**Adaptive responses**' or '**Reflexive responses**'.
- Step 5: Enter your csv data file as an '**Input File**'.
- Step 6: Click on the '**Fit Data**' button.

## Introduction

---

This app allows a user to apply a simplified version of the DIVA model to data collected from auditory perturbation experiments. The data can be from **adaptive** or **reflexive** perturbation paradigms. Adaptive paradigms involve a perturbation that is sustained over many trials, allowing speakers to both correct for the error within a trial but also to adjust their production on subsequent trials. Typically, a single measure is taken per trial and responses are examined across the experiment. Reflexive paradigms involve unexpected perturbations that occur on a subset of trials, where speakers correct for the error within a trial but do not adjust their production on subsequent trials. Responses are measured over the time-course of a given trial and can be averaged across trials or examined on an individual trial basis.

As input, the app takes a spreadsheet (.csv) that has one row per trial of the experiment (adaptive data) or timepoint of the experiment (reflexive data). The first column contains the perturbation value for each trial/timepoint (as a decimal percentage; e.g., a 30% downward shift of the first formant ( $F1$ ) is -0.3), and each subsequent column contains response data. When modeling adaptive data, each column of data is from an individual subject. When modeling reflexive data, each column of data can be from an individual subject (averaged over multiple trials) or from individual trials. The software finds optimal fit values for DIVA model parameters. The model's fit to the data is displayed and statistics for the fit are provided. The resulting parameter values provide a mechanistic interpretation of the experimental data.

The adaptive model has three model parameters: an auditory feedback control gain, a somatosensory feedback control gain, and a trial-to-trial learning (adaptation) rate.

The reflexive model has four model parameters: an auditory feedback control gain, an auditory feedback control delay, a somatosensory feedback control gain, and an auditory velocity control gain.

The **model parameters** capture the following aspects of speech motor control :

- Auditory feedback control gain ( $\alpha_A$ ): When an auditory feedback perturbation is applied, the auditory feedback controller detects a difference between the position of the target sound and the actual production, and attempts to compensate for the perturbation. A higher  $\alpha_A$  leads to a higher compensatory response to an auditory perturbation. This parameter is relevant for both adaptive and reflexive paradigms.

- Somatosensory feedback control gain ( $\alpha_S$ ): When an auditory feedback perturbation is applied, the somatosensory feedback controller attempts to keep the vocal tract in the normal somatosensory configuration for the sound, thus partially counteracting the compensatory response. A higher  $\alpha_S$  leads to a decrease in the compensatory response to an auditory perturbation. This parameter is relevant for both adaptive and reflexive paradigms.
- Trial-to-trial feedforward learning rate ( $\lambda_{FF}$ ): When an auditory feedback perturbation is sustained over a number of trials, the feedforward command for the next trial is updated by some fraction of the feedback-based corrective command for the current trial. A higher  $\lambda_{FF}$  leads to a larger fraction of the corrective command being added to the feedforward command for the next trial. This parameter applies only to adaptive paradigms.
- Auditory feedback control delay ( $\tau_A$ , in seconds): When an auditory feedback perturbation is applied, it takes the speech motor system time to detect and correct for the error. A higher  $\tau_A$  indicates a longer time taken by the auditory feedback controller to initiate the compensatory response to an auditory perturbation. This parameter is relevant only to reflexive paradigms.
- Auditory velocity control gain ( $\alpha_{AV}$ ): When an auditory feedback perturbation is applied, the auditory feedback controller detects a difference between the velocity of the target sound and the actual production, and attempts to compensate for this difference. A higher  $\alpha_{AV}$  leads to a higher compensatory response for velocity errors. This parameter is relevant only to reflexive paradigms.

## Installation

---

Installer files are provided for Windows, Mac, and Linux.

### Windows

1. Download SimpleDIVA\_WinInstaller, unzip, and click to run.
2. Click **'Yes'** to give SimpleDIVA permission to install.
3. Click **'Next >'**.
4. Choose installation folder for SimpleDIVA; the default is to save the app to C:\Program Files in a subfolder called 'GuentherLab'. You can also opt to add a shortcut to the desktop. Click **'Next >'**.
5. Choose installation folder for MATLAB Runtime 2020b; the default is C:\Program Files in the MATLAB subfolder. If Runtime 2020b already exists in this folder, you will not have to choose an installation folder and Runtime will not be installed again. Click **'Next >'**.
6. Click **'Install >'**. Note that if Runtime 2020b needs to be installed, the installation will take a few minutes. Otherwise installation should complete in less than 20 seconds. If there are issues installing MATLAB Runtime through the installer, you can install MATLAB Runtime 2020b manually from the [mathworks website](https://www.mathworks.com/help/matlab/install/installing_matlab_runtime.html).
7. Click **'Finish'**.
8. Go to the folder where you installed SimpleDIVA. Inside SimpleDIVA, there are four subfolders (appdata, application, sys, and uninstall).
9. Go to the **'application'** subfolder and move 'example\_data.csv' and 'example\_data2.csv' to another location on your computer outside of C:\Program Files (e.g., Desktop).

## Mac

1. Download SimpleDIVA\_MacInstaller, unzip, and click to run. You may get a warning that says the installer can't be opened because it is from an unidentified developer. If this happens, right-click on the installer, and click 'open' and then click 'open' again in the pop-up window.
2. Provide your user name and password to allow java to make changes.
3. Click '**Next >**'.
4. Choose installation folder for SimpleDIVA; the default is to save the app to the Applications folder in a subfolder called 'GuentherLab'. Click '**Next >**'.
5. Choose installation folder for MATLAB Runtime 2020b; the default is the Applications folder in the MATLAB subfolder. If Runtime 2020b already exists in this folder, it will not be installed again. Click '**Next >**'.
6. Click '**Install >**'. Note that if Runtime 2020b needs to be installed, the installation will take a few minutes. Otherwise installation should complete in less than 20 seconds. If there are issues installing MATLAB Runtime through the SimpleDIVA\_MacInstaller, you can install MATLAB Runtime 2020b manually from the [mathworks website](#).
7. Click '**Finish**'.

## Linux

Note: The Linux installer has not been fully tested due to limited access to a Linux machine.

1. Download SimpleDIVA\_Linux, go to the download folder, and unzip.
2. Run the following two commands:

```
chmod +x SimpleDIVA_LinuxInstaller.install
```

```
./SimpleDIVA_LinuxInstaller.install
```

3. Follow the command line prompts to complete installation.
4. If the automatic installation (steps 2-3) does not work, please refer to the manual installation instructions in the readme.txt file in the SimpleDIVA\_Linux directory.

## Examples

---

We have included three example data files to get started with the SimpleDIVA app:

- example\_adaptive\_data1.csv contains data from an adaptive experiment and has 11 columns: the first contains the perturbation values per trial and columns 2-11 contain F1 data for 10 different subjects
- example\_adaptive\_data2.csv contains data from an adaptive experiment and has 2 columns: the first contains the perturbation values per trial and column 2 contain F1 data for 1 subject
- example\_reflexive\_data.csv contains data from a reflexive experiment and has 19 columns: the first contains the perturbation values per timepoint (with a 5 ms-interval between timepoints), and columns 2-19 contain f0 data for 18 different subjects

Here we demonstrate how to apply SimpleDIVA to these datasets.

## Individual data or mean data across subjects

1. Go to the folder where you installed SimpleDIVA.
2. Go to the '**application**' subfolder and click on '**SimpleDIVA**' to launch the app.
3. Under '**Experiment Type**', select '**Adaptive responses**' or '**Reflexive responses**', depending on whether you are fitting the example adaptive or reflexive data file.
4. Under '**Input file(s)**', click '**Browse**' and then select the example\_adaptive\_data1.csv or example\_reflexive\_data.csv. The file path and file name will be shown in the top box, and underneath you will see information about the data (number of subjects, number of time points, time points until perturbation onset, and the maximum perturbation magnitude). This information can be used as a sanity check that the correct file was selected and loaded as expected. You can edit the '**Output file**' name to be something more meaningful than 'Default\_Out1'.
5. Under '**Model Parameters**', you have the option to **fix** or **fit** the default model parameters (auditory feedback control gain ( $\alpha_A$ ), somatosensory feedback control gain ( $\alpha_S$ ), and trial-to-trial feedforward learning rate ( $\lambda_{FF}$ ) for adaptive data; auditory feedback control gain ( $\alpha_A$ ), auditory feedback control delay ( $\tau_A$ ), somatosensory feedback control gain ( $\alpha_S$ ), and auditory velocity control gain ( $\alpha_{AV}$ ) for reflexive data. Fixing the parameters means setting them to a pre-determined value rather than having the model find an optimal value, whereas fitting the parameters means having the model find optimal values. Generally, we want to fit the default parameters so that the model will estimate the contribution of the parameters to the data.

For reflexive data simulations, the parameter  $\alpha_{AV}$  is set to 'fixed' at a value of 0 rather than 'fitting' by default. This is because in our testing we found that the addition of this parameter, on average, did not improve the model fit as measured by a corrected Akaike Information Criterion (cAIC) value. You may wish to compare model fits both with and without  $\alpha_{AV}$ . The  $\alpha_{AV}$  parameter adds a term to the model equation that resists perceived  $f_0$  "velocity errors". Since the model assumes participants are trying to maintain a constant  $f_0$ , this term has the effect of counteracting any perceived changes in  $f_0$  and can provide better fits for two-component response patterns, eg., perturbation responses that have an initial rise followed by a plateau or dip and then a second rise. (Note that the model ignores the large  $f_0$  velocities that occur at onset of a pitch perturbation, as including these velocities generally produced poor fits. This corresponds roughly to the assumption that these velocities are unnaturally fast and therefore ignored by the auditory system.) If  $\alpha_{AV}$  is included as a fitting parameter, the model assumes a delay in the auditory velocity error response that is equal in duration to the auditory feedback control delay,  $\tau_A$  unless the advanced option  $\tau_{AV}$  is utilized (see [Advanced Options](#)).

Additional scenarios requiring fixed parameters are explained in the [FAQ](#) section.

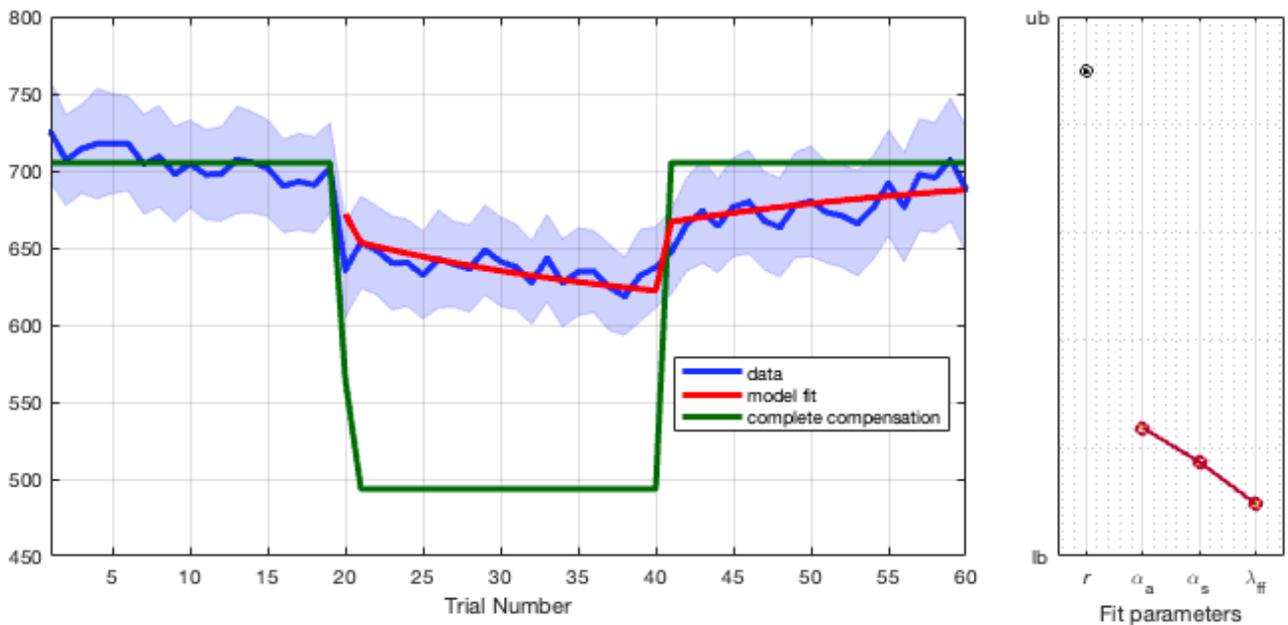
For each fitting parameter, you can also set **lower (LB)** and **upper bounds (UB)**. These correspond to the minimum and maximum values that the model can solve for. As a default (**recommended**), the lower and upper bounds for the parameters are set to 0 and 1, respectively, as this covers the expected range for a neurotypical population. Take the auditory feedback control gain ( $\alpha_A$ ) as an example. An  $\alpha_A$  of less than 0 would mean that a subject responds in the direction of the perturbation instead of opposing it (exacerbating instead of correcting for auditory errors), while an  $\alpha_A$  of greater than 1 would mean a subject overcompensates for auditory errors. Neither of these scenarios is likely true of a neurotypical subject over the long run, but session-to-session and trial-to-trial variability may lead to a dataset that would lead to optimal parameter values that are outside the 0 to 1 range. You

may wish to adjust these ranges to fit clinical populations or to fit individual subjects with highly variable performance measures. However, in the latter case, we recommend instead using **regularization** (described under [Advanced Options](#)), which will seek a solution that maintains reasonable values for parameters, even for subjects with noisy data.

- If modeling reflexive responses, there will be an additional input under 'Model Parameters' for '**Time Step (sec)**'. This corresponds to the time step/interval between data points in the file. For the 'example\_reflexive\_data.csv', set the time step to 0.005.
- Under '**Options**', select '**Fit mean data**'. Selecting this option means that the model will be fit to the mean of all data columns in the .csv file.

The **model equations** are also provided. For concreteness, we have filled F1 into the equations for the adaptive model and f0 into the equations for the reflexive model (F1 is more commonly perturbed in adaptive experiments and vice versa for f0). F1/f0 can be substituted for any auditory dimension being studied (e.g., second formant).

- Click '**Fit Data**'. The model results will appear in the lower section of the app. A 2-panel figures provides a graphical representation of the results:



The left panel figure shows the experimental data (blue) and the model fit (red) as a time series over the number of trials in the experiment. The y-axis units are the same as the input data (F1 (Hz) for example\_adaptive\_data1.csv).

The right panel figure shows the model fit parameters: a Pearson's correlation ( $r$ ) describes the relationship between the data and model fit, and the optimized parameters are provided for all fitting parameters. The y-axis limits range from 0-1 for  $r$  and are scaled based on the lower and upper bounds for the optimized parameters.

The final results shown in the app are the quantitative values for the model fit parameters:

$\alpha_A$	0.2378	$r$	0.8986
$\alpha_S$	0.1744	RMSE	1.292e-01
$\lambda_{FF}$	0.0964	cAIC	-1.240e+00
$F1_T$	705.2602		

The values for the fitting parameters and  $r$  correspond to those plotted in the right panel figure. Additional values are provided for the target F1 or f0 ( $F1_T/F0_T$ ), Root Mean Square Error (RMSE), and cAIC.

- $F1_T/F0_T$ : The target F1 or f0 used for fitting the model. By default, this value is equal to the mean of F1 or f0 during the baseline phase of the experimental data. It is possible to change the target value in the [Advanced Options](#).
- RMSE: A normalized value that captures how well the model fits the data.
- cAIC: A normalized value that captures how well the model fits the data while accounting for the number of fitting parameters. This value is further corrected to account for the temporal dependence between datapoints as well as the number of subjects/samples in a dataset, allowing comparison not only across models but across different datasets.

The RMSE and cAIC measures are described in more detail in the **output files** section below.

9. Under '**Options**', there are four further options that affect the fitting of the data and the left panel figure.

As mentioned above, when '**Fit mean data**' is selected, the model is fit to the mean of all data columns in the .csv file. When 'Fit mean data' is not selected, the model is fit to each data column separately. If there is only a single column of data (as in adaptive\_example\_data2.csv), selecting this option will have no effect on the model output, but it is best to keep 'Fit mean data' selected so that the quantitative values for the model fit parameters are shown. If there are multiple data columns, the plots corresponding to each column will appear one by one in the lower section of the app. Each figure will be saved as before to the folder where you selected the input data from (the output file name will be appended with a # based on the data column).

When '**Include full compensation plot**' is selected, the plot will include a green line showing what full compensation would look like; that is, if F1 was increased by 30%, full compensation would be a 30% decrease in F1. We rarely observe full compensation - formant adaptation experiments tend to show about 30% compensation, while pitch adaptation experiments show about 50% compensation.

When '**Include SEM of data**' is selected, the standard error of the mean of the experimental data across data columns will be shown with blue shading in the left panel figure. If there is only a single column of data, standard error cannot be calculated and selecting this option will have no effect on the figure.

When '**Include standard error of model fit**' is selected, the standard error of 10 iterations of the model-fitting procedure will be shown in red shading in the left panel figure.

We advise against changing the parameters available under '[Advanced Options](#)' (with the exception of **regularization** when fitting noisy individual subject data) unless you are familiar with those aspects of the model.

10. In addition to the display of results in the app, four **output files** are saved to the folder where you selected the input data from:

- **[Output file name]\_stats.txt**: Records the file path/ file name of the input file, fit statistics, and the resulting optimized model parameters. Note that all of the fit statistics are normalized by the number of data columns in a dataset.

```
Read 10 timeseries with 60 trials/timepoints from /Users/elainekearney/Repos/SimpleDIVA/sample-data/adaptive_example_data1.csv
Found 20 samples with missing forment data
Calculated mean data
Calculating optimized parameters. Please wait...
| normalized correlation coefficient = 0.898560
Fit normalized rmse (fit to average of all subjects/samples) = 0.014616 (1.46% of baseline)
| normalized aic (fit to average of all subjects/samples) = -5.467088 (threshold = 0.146133, samples = 41.000000)
| normalized caic (fit to average of all subjects/samples) = -5.467088 (threshold = 0.146133, dof = 41.000000)
Fit normalized rmse (fit to individual-subject/sample data) = 0.129250 (12.9% of baseline)
| normalized aic (fit to individual-subject/sample data) = -1.239510 (threshold = 0.014613)
| normalized caic (fit to individual-subject/sample data) = -1.239510 (threshold = 0.014613)
parameters:
| alpha_a = 0.237815319378427
| alpha_s = 0.174383995834586
| lambda_ff = 0.0964024650722982
```

- **Correlation coefficient**: Calculated as a Pearson's correlation ( $r$ ) describing the relationship between the input data and the model fit.
- **RMSE**: The input data are normalized by dividing all data points by the average of the baseline period, such that the baseline average is 1. The model is then fit to the normalized input data, and RMSE is calculated between the resulting model fit and the normalized input data. The percent of baseline indicates the size of the error relative to the mean of the baseline value. When 'Fit mean data' is checked, RMSE can be calculated in two different ways, both of which are output by the SimpleDIVA software: (1) by comparing model output to the mean data trace formed by averaging across data columns (indicated as '**fit to average of all subjects/samples**'), and (2) by comparing the model output to all of the individual data columns separately and summing these errors across the dataset (indicated as '**fit to individual-subject/sample data**'). The former specifies the RMSE for the model's fit to the mean data trace, while the latter indicates how well the model (with parameters fit to the average trace) fits the individual data traces. One could imagine, for example, a situation in which the model produces a good fit to the mean data trace, but due to very high inter-subject variability the model provides much poorer fits to the individual subject/sample data.
- **AIC/cAIC**: The **Akaike Information Criterion, or AIC**, provides a means for comparing models that takes into consideration both the number of free parameters (in the form of a "punishment" term that increases with the number of free parameters) and normalized RMSE when assessing a model's fit. Lower values of AIC denote better models (in the sense of minimum information loss). SimpleDIVA reports both a "raw" AIC value and a corrected value (cAIC), which takes into consideration temporal dependencies inherent in the data. Specifically, this correction uses the Satterthwaite-Welsh approximation from the observed autocorrelation of the data before the onset of the perturbation when estimating the effective degrees of freedom afforded by the dataset. When "Fit mean data" is checked, AIC/cAIC can be calculated in two different ways (paralleling RMSE), both of which are output by the SimpleDIVA software: (1) by comparing model output to the mean data trace formed

by averaging across data columns (indicated as '**fit to average of all subjects/samples**'), and (2) by comparing the model output to all of the data columns separately and summing these errors across the dataset (indicated as '**fit to individual-subject/sample data**').

In the opinion of the authors, the best measure for comparing two competing models fit to population mean data is cAIC (fit to individual-subject/sample data) as this provides the best account of how well the model can fit the individuals in a population using a single set of parameters. If, on the other hand, the goal is to see how well the model can characterize individuals in a population with **individual-specific** parameters (using a data file with one column of data per subject), 'Fit mean data' should be unchecked, and the best model comparison measure will be the mean of the individual-subject/sample cAIC measures reported in the Summary statistics at the end of such a simulation. If there are multiple trials/samples for each subject (using individual-specific data files with one column of data per trial), we recommend fitting each subject's mean data and using cAIC (fit to individual-subject/sample data) as the best measure for comparing models.

For all AIC/cAIC measures, the 'threshold' corresponds to a significance threshold for evaluating the difference between two models; a model whose AIC value is less than another model's AIC by more than this threshold is, with 95% likelihood, the superior model (there is a 20:1 or above ratio between the likelihood of the two models).

- **[Output file name]\_table.csv**: Records the settings used for model fitting.
- **[Output file name].csv**: Records time-series data of the perturbation magnitude and the model fit.
- **[Output file name].fig**: Saves the model figure as a MATLAB .fig file.

## Data from multiple subjects across different experimental paradigms

Another possibility is to fit multiple data files simultaneously with one set of model parameters. You may want to take this approach if you have multiple subjects within a study who completed different protocols or if you want to fit data from multiple studies together (e.g., see Figure 12, Kearney et al., 2020).

The two adaptive example data files have different experimental paradigms. 'adaptive\_example\_data1.csv' has a long ramp phase and a total of 160 trials, whereas 'adaptive\_example\_data2.csv' has a gradual ramp phase and a total of 60 trials. Note that each variation in experimental paradigm requires a different csv file.

1. Under '**Input file(s)**', click '**Browse**' and then select 'adaptive\_example\_data1.csv' and 'adaptive\_example\_data2.csv' (there is no limit on the number of files that can be fit simultaneously, it will simply take a little longer).
2. You can click on the file path/name under 'Input file(s)' to display the info for a given file (number of subjects, number of time points, and the maximum perturbation magnitude).
3. To fit all files, click '**Select All**'. To fit a subset of the files, use ctrl + click (Windows) or command + click (Mac) to select the target files.
4. Select '**Fit mean data**' and then '**Fit data**'.

The left panel figures includes a subplot of the data and model fit for each .csv file (the first file you entered will be on the bottom). The right panel figure shows the model parameters, optimized to fit across all of the files.

Note that  $r$  is calculated across all trials in all files, and is influenced more heavily by datasets with more trials.

## Advanced Options

---

Advanced options are available with the recommendation that they are best left at their default values.

### Additional parameters

**Three additional parameters** can be fitted in the **adaptive model**. When 'fixed', these parameters are set to default values.

- Auditory error nonlinearity ( $\epsilon$ ): Applies a scaling factor that weights smaller perturbations relatively more than larger perturbations. Default value =  $\exp(-1)$ , which is equivalent to no nonlinearity.
- Auditory target learning rate ( $\lambda_{AT}$ ): Determines how much the auditory target is updated on each production. Default value = 0, corresponding to the assumption that the auditory target for a sound does not change appreciably over the course of an experiment.
- Somatosensory target learning rate ( $\lambda_{ST}$ ): Determines how much the somatosensory target is updated on each production. Default value = 0, corresponding to the assumption that the somatosensory target for a sound does not change appreciably over the course of an experiment.

**Two additional parameters** can be fitted in the **reflexive model**. When 'fixed', these parameters are set to default values.

- Somatosensory feedback control delay ( $\tau_S$ ): The delay (s) associated with the somatosensory feedback controller counteracting the compensation. Default value = .05, corresponding to a .05 s delay.
- Auditory velocity control delay ( $\tau_{AV}$ ): Additional delay (s) associated with auditory velocity error (added to  $\tau_a$  to form the total auditory velocity delay). Default value = 0, corresponding to an auditory velocity delay equal to  $\tau_a$ . Negative values of  $\tau_{av}$  indicate faster processing of auditory velocities compared to auditory "positions" such as f0 or F1.

### Regularization

The **regularization** option is useful when fitting individual subject data with high subject-to-subject variability. An indication to use at least a small amount of regularization when fitting individual subjects is when no regularization results in one or more of the parameters' optimal values being at the end of its bounded range.

When choosing this option, a regularization term acts to keep parameter values from drifting too far from "central" values. By default (**recommended**), the model will use the optimized parameter values found for the mean of the dataset as the central values. It is also possible, however, to set your own custom central values.

To use the regularization function:

1. Check '**Enable regularization**'.
2. In the value box, input the regularization factor, which in effect determines how strongly the parameter values are "punished" for straying far away from the central values during optimization. This value is

typically less than 0.1 and defaults to .001. A good regularization factor is one that (a) eliminates most or all cases in which the optimized parameter is at the end of its range, while (b) still allowing for substantial between-subject differences in optimized parameter values.

3. If you wish for the model to use the parameters from the mean data fit as the central values for regularization, make sure that the '**Use parameters from mean data fit for regularization**' option is checked.
4. Otherwise, uncheck '**Use parameters from mean data fit for regularization**' and switch back to the '**Regular Options**'. The value boxes for each fitting parameter will now be editable. These values are the central values that will be used in regularization.

## Auditory target estimation

In a typical auditory perturbation experiment, the target value of an auditory parameter such as  $f_0$  or  $F_1$  (i.e., the value the speaker is trying to achieve) is estimated by averaging the measured value of the parameter during a pre-perturbation baseline period. The SimpleDIVA model equates the baseline period with all data points that precede the onset of perturbation. By default, the advanced parameter '**Use baseline average as  $F_{1T}/f_{0T}$** ' is checked, and the target is set to the mean of  $F_1$  or  $f_0$  during the baseline phase of the experimental data. When unchecked (**not typically recommended**), a value box for ' $F_{1T}/f_{0T}$ ' becomes active, and you can manually enter a value a target value in normalized units; that is, a value of 1 indicates the baseline average of the parameter, whereas 1.1 corresponds to a target that is 10% larger than the baseline average.

The SimpleDIVA code provides another alternative for estimating the auditory target, which is to find the optimal value of the target (restricted to the range of  $\pm 5\%$  of the baseline average) for minimizing RMSE. This option can be chosen by checking '**Find optimal  $F_{1T}/f_{0T}$** '. Note that this increases the number of fitting parameters by 1, which will lead to longer simulation times and may adversely affect cAIC. Use of this option is only recommended if you are concerned that there are not enough baseline data points to accurately estimate the auditory target through averaging.

## Baseline seeding

There are two options for the model's behavior during the baseline period prior to perturbation onset. By default, the advanced parameter '**Seed model with data**' is checked, and the model uses the trial-by-trial data points from the baseline period up to the value specified in the 'Seed duration (# trials)' or 'Seed duration (sec)' box for adaptive and reflexive data, respectively. This allows the model to consider the time course of the auditory parameter prior to perturbation onset when generating its output, in keeping with studies indicating that a subject's response to a perturbation is affected by what their auditory output was at perturbation onset (Franken et al., 2018). The seed duration is automatically set to the onset of the perturbation for a given file, but can be manually overwritten. For example, if modeling data from a control condition without any perturbation, it may be necessary to change the seed duration. When unchecked, the model output for the baseline period is set to  $F_{1T}/f_{0T}$ .

Regardless of which baseline option is chosen, the RMSE and cAIC calculations ignore the baseline period and utilize only data points after perturbation onset.

# Uninstall SimpleDIVA

---

You can uninstall the SimpleDIVA app at any time. Follow the instructions below for the platform you are using.

## Windows

1. Go to the folder where you installed SimpleDIVA.
2. Go to the '**uninstall\bin**' subfolder and click on '**uninstall.exe**' to uninstall the app.

## Mac

1. Go to the folder where you installed SimpleDIVA and simply delete the folder.

## Linux

1. Go to the folder where you installed SimpleDIVA
2. Run the following command:

```
rm SimpleDIVA
```

## FAQ

---

### Can I fit blocked data as well as individual trial data?

Yes! For some experiments, it makes sense to calculate the mean value over every  $N$  trials in order to reduce the variability in the experimental data. For example, if the stimuli consisted of 3 words that were randomly presented in blocks of 3, it would be appropriate to estimate the mean of every 3 trials. Similarly, if only one stimulus was used throughout the experiment, it would be appropriate to again estimate the mean of every 3 trials. If, however, multiple stimuli were used but were not presented in a blocked fashion, it would be best to retain the individual trial values. You would need to block the data (take the mean over  $N$  trials) when preparing the csv file for input into SimpleDIVA. No other changes are required when interacting with the SimpleDIVA app.

### When would I want to fix the model parameters? And what values do I use?

It's possible to use optimized model parameters from one experimental condition (let's call it condition1) to predict the model fit to another experimental condition (condition2). To do this, first fit condition1.csv as described previously. When fitting condition2.csv (again as Input file 1), select 'Fixed' for all three model parameters. In the 'Value' boxes, insert the optimized model parameters from condition1.csv. Remember that these values have been saved to an output file called [Output file name]\_stats.txt. The model will then use these values when fitting the data and they will remain unchanged in the output. The resulting  $r$  and  $RMSE$  values will reflect how well the parameters from condition1 predicted the model fit in condition2.

## Can I fit data from an experiment that used masking noise?

Yes! Masking noise is sometimes used during the hold phase of experiments to assess adaptation in the absence of auditory feedback. To indicate which trials have masking noise, replace the perturbation value in column 1 of the csv file with 'NaN' (not a number). For the noise-masked trials, no error will be detected between the auditory target and auditory feedback but somatosensory feedback errors will still be detected and corrected.

## What about missing data?

Missing data points can be handled in the model. Either leave the data point blank or replace it with 'NaN'.

## Where can I find the latest version of the app?

<http://sites.bu.edu/guentherlab/software/simplediva-app>

## How do I cite the software?

For the adaptive model, please cite our paper in *Frontiers in Psychology*:

Kearney, E., Nieto-Castañón, A., Weerathunge, H. R., Falsini, R., Daliri, A., Abur, D., Ballard, K. J., Chang, S-E., Chao, S-C., Heller Murray, E. S., Scott, T. L., & Guenther, F. H. (2020). A simple 3-parameter model for examining adaptation in speech and voice production. *Frontiers in Psychology*, *10*, 2995. <https://doi.org/10.3389/fpsyg.2019.02995>

For the reflexive model, we have a manuscript under review. In the meantime, please cite the software and check the website for an updated citation.

Kearney, E., Nieto-Castañón, A., Falsini, R., Daliri, A., Heller Murray, E. S., Smith, D. J., & Guenther, F. H. (2022). *SimpleDIVA* (Version 2.0) [Computer software]. <http://sites.bu.edu/guentherlab/software/simplediva-app>

## References

---

Franken, M. K., Acheson, D. J., McQueen, J. M., Hagoort, P., & Eisner, F. (2018). Opposing and following responses in sensorimotor speech control: Why responses go both ways. *Psychonomic Bulletin & Review*, *25*(4), 1458-1467.

## License

---

[MIT](#)

# Advanced Information for BU Speech Lab Members

## Parallelizing simulations on the shared computer cluster (SCC)

Parallelization is available when running individual subject/trial data (`fit_mean_data = 0`) or when running cross-validation. To take advantage of the improved processing time, you first need to add and save the path to the SimpleDIVA function. Replace `[filepath]` with the path to your clone of the SimpleDIVA GitHub repo.

```
addpath [filepath]/SimpleDIVA.m
savepath ~/matlab/pathdef.m
```

Note that figures cannot be generated/saved when using parallelization, however they can be generated posthoc using the saved stats structure.

You may find that you run into issues with using too many resources on the SCC and having your session "reaped"/terminated as a result. To avoid this from happening, you can force MATLAB to use only a single computational thread at a time. When launching MATLAB from the terminal, use the following command:

```
matlab -singleCompThread
```

One additional factor to consider when using parallelization is that the process generates a number of temporary files stored in your home directory on the SCC (`~.qlog/`). Each user has a quota of files that can be stored in their home directory (max = 200,000), at which point, you may no longer be able to log in to the SCC. To avoid this issue, you can intermittently check your file usage with the following command:

```
quota -s
```

If you are nearing your quota, you can delete the `.qlog` directory (just make sure that your simulations are finished running). This command recursively (`r`; iterates through directories and subdirectories) and forcefully (`f`; does not ask permission) deletes the contents of `.qlog`, while verbosely (`v`) telling you what it's doing.

```
rm -rfv .qlog
```

You will also see a warning on SCC-on-demand if you have used 95% of your quota. For more information on file storage and quotas on the SCC, see the [BU help page](#).