Appendix A

The New Pulsar Generator (nuPG) User Manual

A.1 Installation

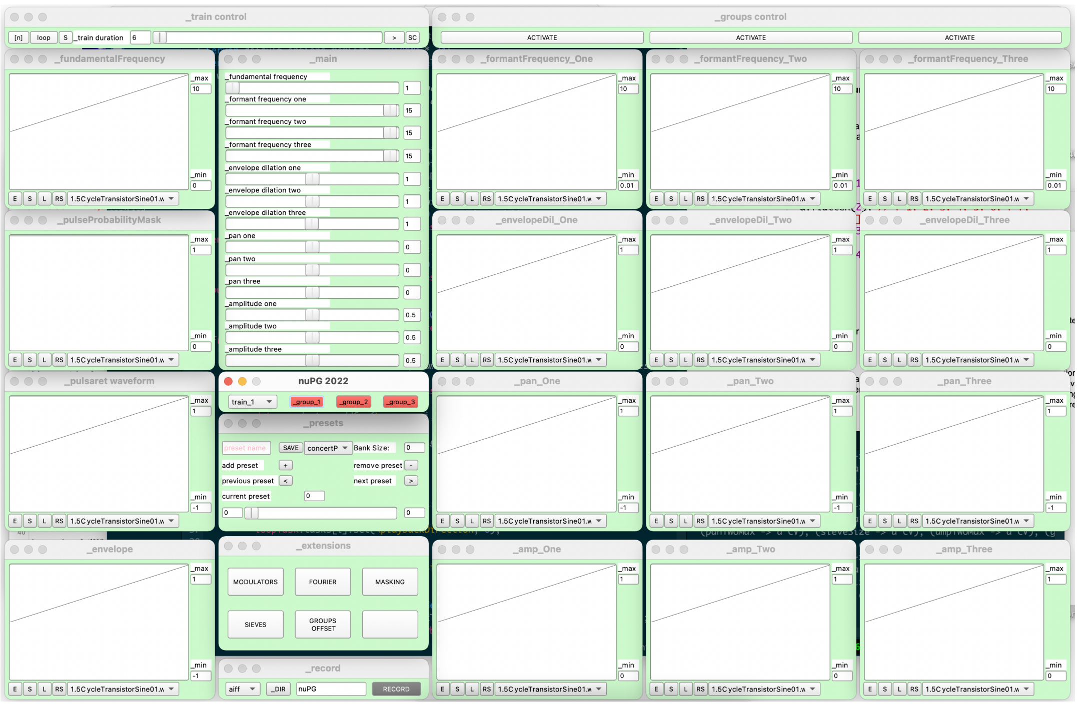

FIGURE A.1: A global view of the New Pulsar Generator (nuPG) standalone program. The majority of parameters are controlled through a table editor. There are sixteen table editors ('fundamental frequency', 'pulse probability masking', 'pulsaret waveform', 'envelope', 'formant frequency one', 'envelope dilation one', 'panning one', 'amplitude one', 'formant frequency two', 'envelope dilation two', 'panning two', 'amplitude two', 'formant frequency three', 'envelope dilation three', 'panning three', 'amplitude three') visible at the startup and three additional ('frequency modulation amount', 'frequency modulation ratio', 'multiparameter modulation') accessible from within MODULATORS extension.

The New Pulsar Generator (nuPG) program comes packaged as a standalone application for the Mac OS system. Three architectures—Intel, M1 and M2—are supported. The program can also be launched from within SuperCollider IDE, running a script **nuPG_startUp2022.scd**. Running the program from the script requires the user to import source code and additional classes into the local SuperCollider directory. See (D) for details.

A.2 Table Editors

The program at the startup displays a set of graphic interface windows (A.1). The majority of synthesis parameters are controlled through a set of table editors. There are sixteen table editors visible at the startup and three additional accessible from the 'MODULATORS' extension. Each table editor (A.2) consists of a table view, a set of functions ('RS' - resize the table, 'L' - load a table from a file, 'S' - save a table to a file, 'E' - open large table editor), a drop-down menu with a precompiled set of tables, and a set of values setting the table's range. The table view display can be drawn on. There are 2048 values on the display. The data from a table can be copied pressing 'C' button on the keyboard and pasted into any table-type object of the program by presssing 'P' on the keyboard. Other keyboard shortcouts inculde 'R' to reverse the data, 'I' to invert it, 'M' for multiplication

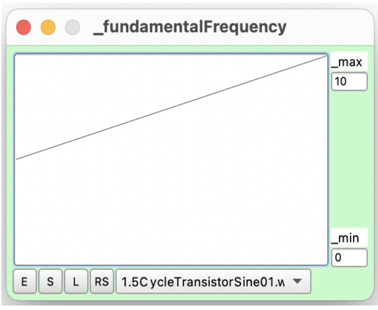

FIGURE A.2: Each table editor consists of a table view, a set of functions ('RS' - resize the table, 'L' - load a table from a file, 'S' - save a table to file, 'E' - open large table editor), a drop-down menu with precompiled tables, and a set of values setting the table's range

Each table editor is supplied with an additional enlarged editor (A.3). Besides providing a finer resolution for table data, the editor gives access to additional methods: 'R' - reverse table data, 'I' - inverse table data along the horizontal axis, 'NM' - multiplication, 'RM' - ring modulation, '←' - shift left, '→' - shift right, '↑' - shift up, '↓' - shift down.

A.3 Train Control

The train playback is controlled via train control GUI (A.4). A set of controls displayed allows to interact with the pulsar train; the '[n]' button starts and stops the train, the 'loop' button activates the looping function across tables, the 'S' button synchronises multiple trains, 'train duration' sets a duration of the train in seconds (available range is 1 to 120), the '>' button changes the direction of the table read, and the 'SC' button opens a scanner editor (A.5). When the pulsar train starts, it only reads values from the 'MAIN' control GUI. To activate the table reading, the 'loop' function needs to be set to 

. The loop function by default loops through 'fundamental frequency' and 'pulseProbabilityMask' tables only. In order to activate one of three groups of 'formant freqency', 'envelope dillation', 'panning' and 'amplitude' tables set 

to 

.

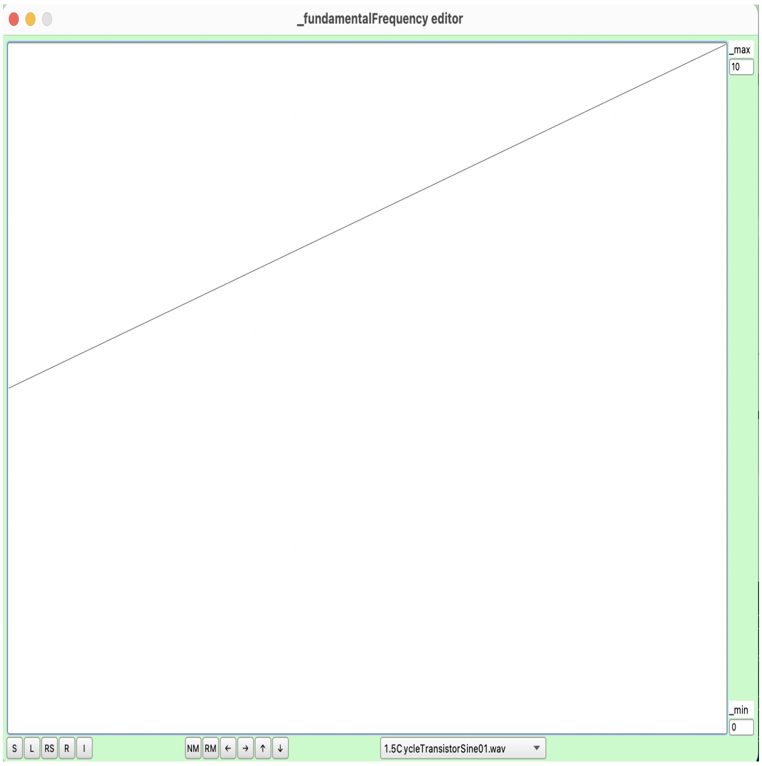

FIGURE A.3: Each large table editor consists of a table view, a set of functions ('RS' - resize the table, 'L' - load a table from a file, 'S' - save a table to file, 'R' - reverse table data, 'I' - inverse table data along the horizontal axis, 'NM' - multiplication, 'RM' - ring modulation, '←' - shift left, '→' - shift right, '↑' - shift up, '↓' - shift down.), a drop-down menu with precompiled tables, and a set of values setting the table's range

The function is accessible via 'GROUPS CONTROL' GUI located at the top right of the screen.

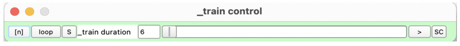

FIGURE A.4: A set of controls displayed allows for interaction with the pulsar train; the '[n]' button starts and stops the train, the 'loop' button activates the looping function across tables, the 'S' button synchronises multiple trains, 'train duration' sets a duration of the train in seconds (available range is 1 to 120), the '>' button changes the direction of the table read, and the 'SC' button opens a scanner editor.

The scanner GUI (A.5) allows reading values from tables according to user-specified path. Holding and dragging a grey bar across reads values from all active tables. The movement can be recorded by pressing 'R' button on the right side of the GUI. The movement is played back automatically after the release of the slider bar. The 'S' button on the left side of the GUI synchronises the scanning function across all program instances. The display to the right of the slider shows a table slot value between 0-2047 (2048).

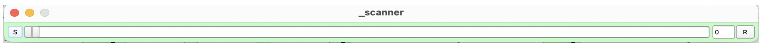

FIGURE A.5: The scanner GUI allows to read values from tables according to the user-specified path. Holding and dragging a grey bar across reads values from all active tables. The movement can be recorded by pressing 'R' button on the right side of the GUI. The movement is played back automatically after the release of the slider bar. The 'S' button on the left side of the GUI synchronises the scanning function across all program instances. The display to the right of the slider shows a table slot value between 0-2047 (2048).

A.4 Main Control

The 'MAIN CONTROL' GUI (A.6) provides a direct access to a set of key synthesis parameters: 'fundamental frequency', 'formant frequency one', formant frequency two', formant frequency three', 'envelope dilation one', 'envelope dilation two', 'envelope dilation three', 'pan one', 'pan two', 'pan three', 'amp one', 'amp two', and 'amp three'. The relationship between the parameters of the 'MAIN' and 'TABLES' GUI is a modulation (A.7). The table minimum and maximum values act as modulation depth. The value from the 'MAIN' GUI gets multiplied by or added to (this is the case for the panning parameter) a value from the table at each index according to to 'train duration' value. For example, with the 'train duration' of 6 seconds and 'MAIN' GUI's 'fundamental frequency' set to 2 cycles per second gets multiplied by a new table value at every 0.0029296875 of a second ($6 \div 2048 = 0.0029296875$).

A.5 Presets

The data of the program can be stored as a preset. The 'PRESETS' GUI (A.8) provides an access to saving, recalling and interpolating preset data. To add a preset, press the '+' button; to remove a preset, press the '-' button. Buttons '<' and '>' move up

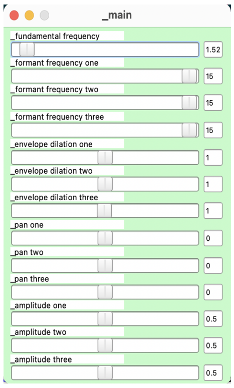

FIGURE A.6: The 'MAIN CONTROL' GUI provides a direct access to a set of key synthesis parameters: 'fundamental frequency', 'formant frequency one', formant frequency two', formant frequency three', 'envelope dillation one', 'envelope dillation two', 'envelope dillation three', 'pan one', 'pan two', 'pan three', 'amp one', 'amp two', and 'amp three'.

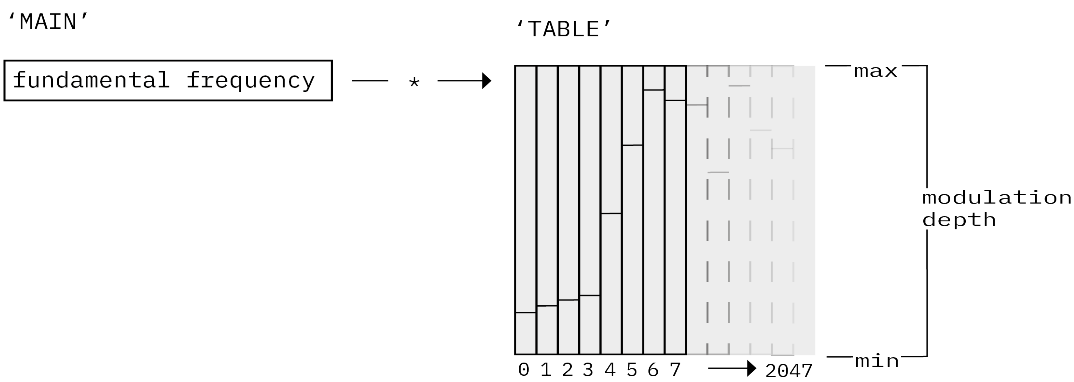

FIGURE A.7: The relationship between the parameters of the 'MAIN' and 'TABLES' GUI is a modulation. The table minimum and maximum values act as modulation depth. The value from the 'MAIN' GUI gets multiplied by or added to (this is the case for the panning parameter) a value from the table at each index according to to 'train duration' value. For example, with the 'train duration' of 6 seconds and 'MAIN' GUI's 'fundamental frequency' set to 2 cycles per second gets multiplied by a new table value at every 0.0029296875 of a second ($6 \div 2048 = 0.0029296875$)

and down across a set of saved presets. Bank Size displays the number of available presets. Presets can be saved as a bank by giving it a name and pressing SAVE button. The new bank appears on the drop-down menu. At the bottom of the GUI user can access preset interpolation functionality. The interpolator requires a value setting for starting preset (current preset) and target preset. By moving the slider from right to left program interpolates between data stored on chosen presets.

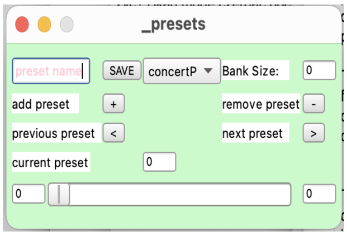

FIGURE A.8: The 'PRESETS' GUI provides an access to saving, recalling and interpolating preset data. To add a preset, press the '+' button; to remove a preset, press the '-' button. Buttons '<' and '>' move up and down across a set of saved presets. Bank Size displays the number of available presets. Presets can be saved as a bank by giving it a name and pressing SAVE button. The new bank appears on the drop-down menu. At the bottom of the GUI user can access preset interpolation functionality. The interpolator requires a value setting for starting preset (current preset) and target preset. By moving the slider from right to left program interpolates between data stored on chosen presets.

A.6 Recording

The output of the program can be recorded using built-in utility tool. The 'RECORD' GUI (A.9) facilitates recording of the audio output of the program. The user can specify the file format (.aiff, .wav, .caf), the directory (default directory on Mac OS is User Music SuperCollider Recordings), and the name of the file (by default files are named 'nuPG' + date + time).

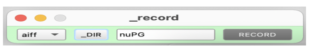

FIGURE A.9: The 'RECORD' GUI facilitates recording of the audio output of the program. The user can specify the file format (.aiff, .wav, .caf), the directory (default directory on Mac OS is User Music SuperCollider Recordings), and the name of the file (by default files are named 'nuPG' + date + time).

A.7 Extensions

The 'EXTENSIONS' GUI (A.10) gives access to additional processes embedded within the nuPG program.

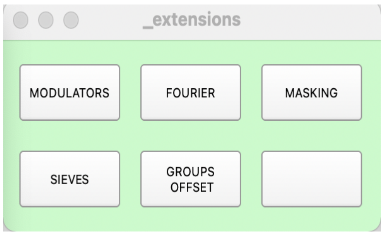

FIGURE A.10: The 'EXTENSIONS' GUI gives access to additional processes embedded within the nuPG program. Extensions include: 'MODULATORS', 'FOURIER', 'MASKING', 'SIEVES' and 'GROUP OFFSET'

A.7.1 Modulators

The 'MODULATORS' (A.11) extension provides two methods for modulation of the nuPG train: rate modulation and multiparameter modulation. The rate modulation applies a modulator to the emission rate of the train. The modulator consists of a sawtooth oscillator fed through a sample and hold function which holds input signal values when triggred. A combination of a modulator with a sample and hold function creates characteristic stepped interlocked patterns. The multiparameter modulation modulates fundamental frequency, formant frequency and amplitude by a dynamic cubic noise generator. The generator outputs polynomially interpolated random values at given rate.

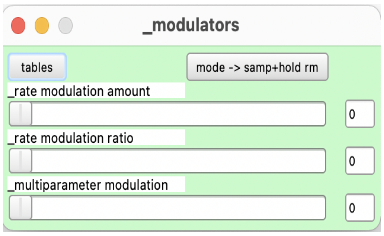

FIGURE A.11: A set of modulation functions (rate modulation and multiparameter modulation) with additional table controls

A.7.2 Fourier

The 'FOURIER' (A.12) editor facilitates a generation of tables through a summation of sinusoidal functions at given amplitude. Each of sixteen bars represents amplitude value from 0 to 1. The data from the editor (A.13) can be copied using 'C' button on the keyboard and pasted into any other table-type object of the program by pressing 'P' on the kayboard.

A.7.3 Masking

The 'MASKING' (A.14) GUI implements three types of pulsar masking: probability, burst and channel. The probability masking omits pulses according to a statistical gate function. When the function receives a trigger, it tosses a coin and either passes the trigger or doesn't. The probability value of 1 equals pulses all the time, and 0 equals no pulses. Values between 0.9 and 0.7 create a stochastically generated intermittent pattern of pulses and silences. The burst masking generates the

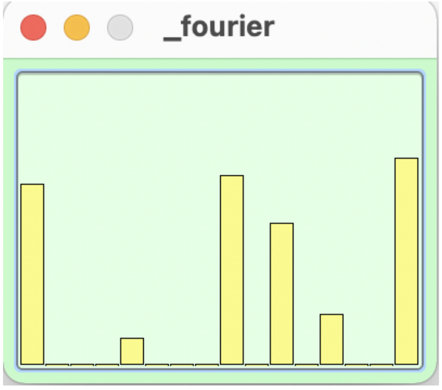

FIGURE A.12: Fourier Editor

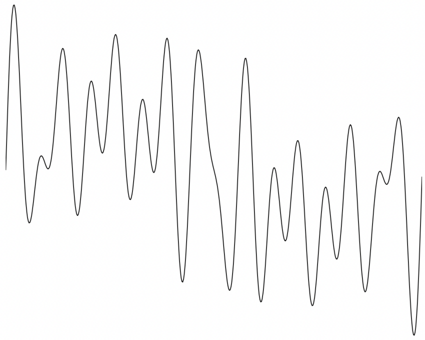

FIGURE A.13: A corresponding table data

mask according to two values: burst and rest. The burst value indicates the number of pulses to preserve, and the rest value sets the number of pulses to mask. The channel-based mask takes one value; the number of pulses to emmit interchangeably between available audio outputs. By default, the output of the nuPG program is set to stereo; the channel masking function adapts to variable outputs.

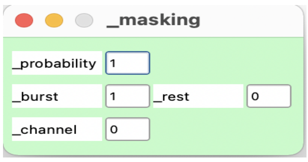

FIGURE A.14: The 'MASKING' GUI implements three types of pulsar masking: probability, burst and channel

A.7.4 Sieves

Sieves are an original implementation within the nuPG program. The GUI (A.15) allows quick access to the definition of new sieves and their combination through logical operators ('|' for union, '&' for section, '–' for symmetric difference, and '-' for difference), and the output as a masking sequence or table data. The sieve as mask functions in two modes: 'sequentialBinaryMask' and 'segmentedBinaryMask'. The 'sequentialBinaryMask' method takes an integer representation (points) of a sieve, replaces each point with one and fills all spaces (slots) that are not occupied with zeros. E.g., Sieve = [3, 7, 11, 15] will be converted to [1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1]. This approach is based on the Python function 'discreteBinaryPad' developed by Christopher Ariza within his *athenaCL*. The 'segmentedBinaryMask' method takes an integer representation (points) of a sieve, gets all the slots from given sieve - this is always one value less than the input list - replaces each integer with a corresponding number of 0 and 1 in a sequence. E.g. input = [2,3,3] -> output [0,0,1,1,1,0,0,0]. This approach is an original development within the nuPG. The technique of masking, when applied to discrete (infrasonic range) pulses creates rhythmic patterns of variable regularity; its application to continuous (audio range) produces amplitude modulation on timbre.

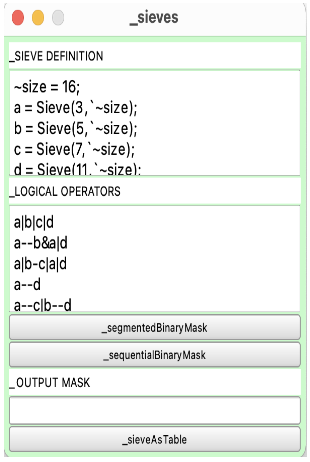

FIGURE A.15: The GUI allows quick access to the definition of new sieves and their combination through logical operators (’|’ for union, ’&’ for section, ’–’ for symmetric difference, and ’-’ for difference), and the output as a masking sequence or table data.

The output of the sieve method as a table generates data which can be pasted (’P’) into any table-data object of the program. The size of the sieve can be made larger for finer resolution of the sieve-generated table.

A.7.5 Groups Offset

The method of group offset is an original development within the nuPG program. The ’GROUP OFFSET’ GUI (A.16) provides access to a function offsetting one of three groups of formant frequency, envelope dilation, panning and amplitude by an offset value between 0 and 1 second.

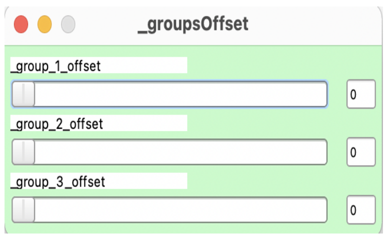

FIGURE A.16: The interface allows to offset of one of three groups of formant frequency, envelope dilation, panning and amplitude by an offset value between 0 and 1 second