Software Defined Visible Light Communication (VLC) Toolbox for GNURadio

Objective:

To create an open-source library for modelling VLC systems within the GNURadio software development toolkit. The idea is to have a variety of blocks in this open-source library which can be used by other researchers to model their own VLC systems in the GNURadio environment.

Acronyms Used:

1. VLC	Visible Light Communication
2. OWC	Optical Wireless Communication
3. E2E	Electrical to Electrical
4. O2O	Optical to Optical
5. O2E	Optical to Electrical
6. E2O	Electrical to Optical
7. DC	Direct Current
8. FOV	Field of View
9. OOK	On-Off Keying
10. PAM	Pulse Amplitude Modulation
11. VPPM	Variable Pulse Position Modulation
12. PWM	Pulse Width Modulation

Blocks Documentation:

Implemented:

1. **Optical Wireless Channel Model:**

The channel model implemented is the IM/DD optical wireless channel model. The DC channel gain in this model is given by the following equation:

$$H_{\mathrm{ij}}\left(\phi_{ij}, \psi_{ij}, d_{ij}\right) = \frac{P_r^{(j)}}{P_t^{(i)}} = \frac{C_T^{(i)} . G_T\left(\phi_{ij}\right). G_R\left(\psi_{ij}\right). C_R^{(j)}}{d_{ij}^2}$$

The above relation is for i_{th} transmitter and j_{th} receiver.

There are two blocks that simulate this channel model and they differ based on the parameters they require.

Inputs and Outputs:

The inputs and outputs for the OWC channel blocks are electrical signals. The input(s) represents an electrical signal from a VLC transmitter(s) whereas the output(s) represents an electrical signal from a VLC receiver(s).

However, these blocks can be used as optical or hybrid (electrical and optical) channels as well. This means that the blocks can also simulate optical signals as input and output or can accept electrical signals at one end with optical signals at the other end. In order to do this, the factor used for electrical to optical conversion C_T or optical to electrical conversion C_R must be set accordingly.

Hence, the channel blocks in terms of their inputs and outputs can be used as E2E, O2O, O2E and E2O. One important point to note here is that it is assumed that these conversions operate in the linear conversion region, allowing C_T and C_R to be specified as constant conversion factors.

i. **Variant 1: *OWC_Channel_relative***

Description:

This block relates transmitted electrical signals/optical power from N transmitters to received electrical signals/optical power at M receivers for a given set of NM emission angles, NM acceptance angles, and NM distances. Lambertian emission is assumed and therefore, a set of Lambertian orders are provided for the N transmitters. Other parameters include a set of M photosensor areas, M transmittance functions of optical filters, M refractive indexes and M concentrator field of view angles. Lastly, N E2O conversion factors and M O2E conversion factors are to be specified.

Mathematical Relationship:

The following relations are for i_{th} transmitter and j_{th} receiver.

$$G_T(\phi_{ij}) = \frac{m_{ij}+1}{2\pi}\cos^{m_{ij}}(\phi_{ij})$$

$$G_R(\psi_{ij}) = A\,.\,T_s\,.\,g(\psi_{ij})\cos(\psi_{ij}) \qquad where \;\; g(\psi_{ij}) = \frac{{n_j}^2}{\sin^2(\boldsymbol{\psi}_{\mathbf{C_j}})} \quad for\; 0 \le \psi_{ij} \ge \boldsymbol{\psi}_{\mathbf{C_j}}$$

$$g(\psi_{ij}) = 0 \qquad else$$

${C_T}^{(i)}$ *and* $C_R^{(j)}$ are specified by the user

$$P_R^{(j)} = \sum_i (P_T^{(i)}.H_{ij})$$

Parameters:

i. **Num_transmitters** – Number of optical transmitters given as a positive integer.

ii. **Num_receivers –** Number of optical receivers given as a positive integer.

iii. **Emission_angle_array –** A single dimensional array (vector) of floating points which contains the emission angles of all the transmitters with respect to all the receivers in series. The values in the array must be listed such that the all the emission angles for receiver # 1 should be provided first starting from transmitter # 1. In other words, all the emission angles from transmitter # 1 to transmitter # N must be listed in a sequence with respect to the receiver # 1. Then, this should be repeated for all the M number of receivers.

iv. **Acceptance_angle_array –** A single dimensional array (vector) of floating points which contains the acceptance angles of all the receivers with respect to all the transmitters in series. The values in the array must be listed such that the all the acceptance angles of receiver # 1 should be provided first starting from transmitter # 1. In other words, all the acceptance angles of receiver # 1 with respect to transmitter # 1 to transmitter # N must be listed in a sequence. Then, this should be repeated for all the M number of receivers.

v. **Distance_array –** A single dimensional array (vector) of floating points which contains the distance between every transmitter-receiver pair (possible combinations). The values in the array must be listed such that all the distances between receiver # 1 to all the transmitters should be provided first starting from transmitter # 1. In other words, all the distances from receiver # 1 to transmitter # 1 till transmitter # N must be listed in a sequence. Then, this should be repeated for all the M number of receivers.

vi. **Lambertian_order_array –** A single dimensional array (vector) of floating points which contains the Lambertian order of all the optical transmitters in a sequence starting from transmitter # 1.

vii. **Photosensor_area_array –** A single dimensional array (vector) of floating points which contains the photosensor area of all the optical receivers in a sequence starting from receiver # 1.

viii. **Optical_filter_transmittance_array -** A single dimensional array (vector) of floating points which contains the transmittance function of the optical filter of all the optical receivers in a sequence starting from receiver # 1.

ix. **Refractive_index_array** - A single dimensional array (vector) of floating points which contains the refractive indexes of all the optical receivers in a sequence starting from receiver # 1.

x. **Concentrator_FOV_array** - A single dimensional array (vector) of floating points which contains the concentrator’s FOV angles of all the optical receivers in a sequence starting from receiver # 1.

xi. **E2O_conversion_factor_array** - A single dimensional array (vector) of floating points which contains the electrical to optical conversion factors of all the optical transmitters in a sequence starting from transmitter # 1.

xii. **O2E_conversion_factor_array** - A single dimensional array (vector) of floating points which contains the optical to electrical conversion factors of all the optical receivers in a sequence starting from receiver # 1.

Parameter to variable mapping:

1. num_Transmitters	N
2. num_Receivers	M
3. Emission_angle_array	$[\phi_1^1, \dots \dots, \phi_N^1, \dots \dots, \phi_1^M, \dots \dots, \phi_N^M]$
4. Acceptance_angle_array	$[\psi_1^1, \dots \dots, \psi_N^1, \dots \dots, \psi_1^M, \dots \dots, \psi_N^M]$
5. Distance_array	$[d_1^1, \dots \dots, d_N^1, \dots \dots, d_1^M, \dots \dots, d_N^M]$
6. Lambertian_order_array	$[m_1, \dots \dots, m_N]$
7. Photosensor_area_array	$[A^1, \dots \dots, A^M]$
8. Optical_filter_transmittance_array	$[T_s^1, \dots \dots, T_s^M]$
9. Refractive_index_array	$[n^1, \dots \dots, n^M]$
10. Concentrator_FOV_array	$[\boldsymbol{\psi}_{\mathbf{c}}^{\mathbf{1}}, \dots \dots, \boldsymbol{\psi}_{\mathbf{c}}^{\mathbf{M}}]$
11. E2O_conversion_factor_array	$[C_{T_1}, \dots \dots, C_{T_N}]$
12. O2E_conversion_factor_array	$[C_R^1, \dots \dots, C_R^M]$

*In the above table, the subscript numbering denotes the number of the transmitter whereas the superscript numbering denotes the number of the receiver.

Example:

The functionality of this block is demonstrated using an example named OWC_Channel_relative_example provided in the examples folder. The example considers two transmitters and two receivers where simple waveforms have been transmitted through the channel block.

ii. Variant 2: *OWC_Channel_absolute*

Description:

This block relates transmitted electrical signals/optical power from N transmitters to received electrical signals/optical power at M receivers for a given set of N 3d location coordinates of the transmitters, N 3d orientation vectors of the transmitters, M 3d location coordinates of the receivers and M 3d orientation vectors of the receivers. Lambertian emission is assumed and therefore, a set of Lambertian orders are provided for the N transmitters. Other parameters include a set of M photosensor areas, M transmittance functions of optical filters, M refractive indexes and M concentrator field of view angles. Lastly, N E2O conversion factors and M O2E conversion factors are to be specified.

Mathematical Relationship:

The following relations are for i_{th} transmitter and j_{th} receiver.

$$d_{ij} = \sqrt{\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2 + \left(z_j - z_i\right)^2}$$

$$\phi_{ij} = \cos^{-1}\left(\frac{\overrightarrow{u_i}\,.\,\overrightarrow{v_{ij}}}{\|\overrightarrow{u_i}\|\|\overrightarrow{v_{ij}}\|}\right) \qquad where\ \overrightarrow{v_{ij}} = \left(x_j - x_i\right)\hat{a}_x + \left(y_j - y_i\right)\hat{a}_y + \left(z_j - z_i\right)\hat{a}_z$$

$$\psi_{ij} = \cos^{-1}\left(\frac{\overrightarrow{u_j}\,.\,\overrightarrow{v_{ji}}}{\|\overrightarrow{u_j}\|\|\overrightarrow{v_{ji}}\|}\right) \qquad where\ \overrightarrow{v_{ji}} = \left(x_i - x_j\right)\hat{a}_x + \left(y_i - y_j\right)\hat{a}_y + \left(z_i - z_j\right)\hat{a}_z$$

$$G_T\left(\phi_{ij}\right) = \frac{m_{ij} + 1}{2\pi}\cos^{m_{ij}}(\phi_{ij})$$

$$G_R\left(\psi_{ij}\right) = A\,.\,T_s\,.\;g\left(\psi_{ij}\right)\cos\left(\psi_{ij}\right) \qquad \text{where} \quad g\left(\psi_{ij}\right) = \frac{{n_j}^2}{\sin^2(\boldsymbol{\psi}_{\mathbf{C_j}})} \quad for\ 0 \le \psi_{ij} \ge \boldsymbol{\psi}_{\mathbf{C_j}}$$

$$g\left(\psi_{ij}\right) = 0 \qquad else$$

${C_T}^{(i)}\ and\ C_R^{(j)}$ are specified by the user

$$P_R^{(j)} = \sum_i (P_T^{(i)}.\,H_{ij})$$

Parameters:

i. **Num_transmitters –** Number of optical transmitters given as a positive integer.

ii. **Num_receivers –** Number of optical receivers given as a positive integer.

iii. **Tx_coordinates_array** – A single dimensional array (vector) of floating points which contains the cartesian coordinates (XYZ) of all the optical transmitters in a sequence starting from transmitter # 1. The XYZ values are provided as separate values in a series.

iv. **Tx_orientation_array** – A single dimensional array (vector) of floating points which contains the cartesian coordinates (XYZ) of the orientation (direction) vectors of all the optical transmitters in a sequence starting from transmitter # 1. The orientations are defined with respect to the origin. The XYZ values are provided as separate values in a series.

v. **Rx_coordinates_array** – A single dimensional array (vector) of floating points which contains the cartesian coordinates (XYZ) of all the optical receivers in a sequence starting from receiver # 1. The XYZ values are provided as separate values in a series.

vi. **Rx_orientation_array** – A single dimensional array (vector) of floating points which contains the cartesian coordinates (XYZ) of the orientation (direction) vectors of all the optical receivers in a sequence starting from receiver # 1. The orientations are defined with respect to the origin. The XYZ values are provided as separate values in a series.

vii. **Lambertian_order_array –** A single dimensional array (vector) of floating points which contains the Lambertian order of all the optical transmitters in a sequence starting from transmitter # 1.

viii. **Photosensor_area_array –** A single dimensional array (vector) of floating points which contains the photosensor area of all the optical receivers in a sequence starting from receiver # 1.

ix. **Optical_filter_transmittance_array -** A single dimensional array (vector) of floating points which contains the transmittance function of the optical filter of all the optical receivers in a sequence starting from receiver # 1.

x. **Refractive_index_array** - A single dimensional array (vector) of floating points which contains the refractive indexes of all the optical receivers in a sequence starting from receiver # 1.

xi. **Concentrator_FOV_array** - A single dimensional array (vector) of floating points which contains the concentrator field of view angles of all the optical receivers in a sequence starting from receiver # 1.

xii. **E2O_conversion_factor_array** - A single dimensional array (vector) of floating points which contains the electrical to optical conversion factors of all the optical transmitters in a sequence starting from transmitter # 1.

xiii. **O2E_conversion_factor_array** - A single dimensional array (vector) of floating points which contains the optical to electrical conversion factors of all the optical receivers in a sequence starting from receiver # 1.

Parameter to variable mapping:

1. num_Transmitters	N
2. num_Receivers	M
3. Tx_coordinates_array	$[x_1, y_1, z_1, \dots\dots, x_N, y_N, z_N]$
4. Tx_orientation_array	$[u_{x_1}, u_{y_1}, u_{z_1}, \dots\dots, u_{x_N}, u_{y_N}, u_{z_N}]$
5. Rx_coordinates_array	$[x^1, y^1, z^1, \dots\dots, x^M, y^M, z^M]$
6. Rx_orientation_array	$[u_x^1, u_y^1, u_z^1, \dots\dots, u_x^M, u_y^M, u_z^M]$
7. Lambertian_order_array	$[m_1, \dots\dots, m_N]$
8. Photosensor_area_array	$[A^1, \dots\dots, A^M]$
9. Optical_filter_transmittance_array	$[T_s^1, \dots\dots, T_s^M]$
10. Refractive_index_array	$[n^1, \dots\dots, n^M]$

11. Concentrator_FOV_array	$[\boldsymbol{\psi}_{\mathbf{c}}^{\mathbf{1}}, \ldots \ldots, \boldsymbol{\psi}_{\mathbf{c}}^{\mathbf{M}}]$
12. E2O_conversion_factor_array	$[C_{T_1}, \ldots \ldots, C_{T_N}]$
13. O2E_conversion_factor_array	$[C_R^1, \ldots \ldots, C_R^M]$

*In the above table, the subscript numbering denotes the number of the transmitter whereas the superscript numbering denotes the number of the receiver.

Example:

The functionality of this block is demonstrated using an example named OWC_Channel_absolute_example provided in the examples folder. The example considers two transmitters and two receivers where simple waveforms have been transmitted through the channel block. Also, the constellation sink has been used to visualize the location of the transmitters and the receivers. Due to the limitations of visualizing 3d space in GNURadio, only the xy coordinates have been used to define the locations.

Assumptions:

1. Lambertian emission is assumed for the transmitters.
2. O2E and E2O conversions operate in the linear conversion region, allowing C_T and C_R to be specified as constant conversion factors.
3. The transmittance is constant within the receiver's FOV.
4. Location, rotation and Lambertian order are the dynamic parameters only (can be changed at runtime).

2. OOK Modulator:

In OOK modulation, the data is transmitted as a signal with only two levels which specify the On and Off states. Hence, only a single bit is modulated per symbol where the higher magnitude denote the bit 1 whereas the lower magnitude denote the bit 0.

Inputs and Outputs:

The input to the OOK block is a stream of bits while the output is a continuous two-level electrical signal which is the modulated signal ready for transmission.

There are two blocks that simulate this modulator which differ based on the parameters they require.

i. Variant 1: *OOK_Modulator_one*

Description:

This block relates the incoming floating point stream (0s and 1s representing bits) at a transmitter to the transmitted two-level signal/waveform for the given values of maximum magnitude of the signal, minimum magnitude of the signal and number of output samples per symbol.

Parameters:

i. **Max_magnitude –** The maximum value of the modulated signal or the magnitude of the higher level of the modulated signal given as a floating point.

ii. **Min_magnitude –** The minimum value of the modulated signal or the magnitude of the lower level of the modulated signal given as a floating point.

iii. **Samples_per_symbol –** The number of output samples per each input symbol given as an integer.

ii. **Variant 2: *OOK_Modulator_two***

Description:

This block relates the incoming floating point stream (0s and 1s representing bits) at a transmitter to the transmitted two-level signal/signal for the given values of amplitude of the signal, mean of the signal and number of output samples per symbol.

Parameters:

i. **Amplitude –** The value of the amplitude of the modulated signal.

ii. **Mean –** The mean value of the modulated signal.

iii. **Samples_per_symbol –** The number of output samples per each input symbol given as an integer.

3. **PAM Modulator:**

In PAM modulation, the data is transmitted as a signal with a series of levels specifying the different symbols. The main advantage of PAM is the possibility of modulating multiple bits per symbol which increases the data rate.

The PAM modulator is realized by two blocks. The first one is a symbol mapper which converts the incoming bits into decimal values and the second one maps the decimal values to the corresponding signal level.

Inputs and Outputs:

The input to the PAM block (symbol mapper + modulator) is a stream of floating point stream (0s and 1s representing bits) while the output is a continuous multi-level electrical signal which is the modulated signal ready for transmission.

i. ***binary_to_decimal_mapper***

Description:

This block converts the incoming floating point stream (0s and 1s representing bits) at a transmitter into decimal values for the given value of PAM signal levels (for deciding how many bits are to be mapped to a single decimal value)

Parameters:

i. **Modulation_order –** The number of symbols in the constellation corresponding to the PAM signal levels to be given as an integer value.

ii. ***PAM_Modulator_one***

Description:

This block relates the incoming decimal value at a transmitter to the transmitted multi-level modulated signal for the given values of modulation order, maximum magnitude of the signal, minimum magnitude of the signal and number of output samples per symbol.

Parameters:

i. **Modulation_order –** The number of symbols in the constellation corresponding to the PAM signal levels to be given as an integer value.

ii. **Max_magnitude –** The maximum value of the modulated signal or the magnitude of the higher level of the modulated signal given as a floating point.

iii. **Min_magnitude –** The minimum value of the modulated signal or the magnitude of the lower level of the modulated signal given as a floating point.

iv. **Samples_per_symbol –** The number of output samples per each input symbol given as an integer.

iii. ***PAM_Modulator_two***

Description:

This block relates the incoming decimal value at a transmitter to the transmitted multi-level modulated signal for the given values of modulation order, amplitude of the signal, mean of the signal and number of output samples per symbol.

Parameters:

i. **Modulation_order** – The number of symbols in the constellation corresponding to the PAM signal levels to be given as an integer value.

ii. **Amplitude** – The value of the amplitude of the modulated signal.

iii. **Mean** – The mean value of the modulated signal.

iv. **Samples_per_symbol** – The number of output samples per each input symbol given as an integer.

4. **VPPM Modulator:**

In VPPM modulation, the duty cycle of each symbol is changed (PWM) in order to encode the bits and the symbols are distinguished by the pulse position. VPPM supports illumination with dimming control and communication and therefore, is an ideal modulation scheme for VLC.

Here, we have implemented 2-VPPM meaning that the modulation order is 2 and there are only two defined pulse positions.

Inputs and Outputs:

The input to the VPPM block is a floating point values (0s and 1s representing bits) while the output is a continuous two-level electrical signal which is the VPPM modulated signal ready for transmission.

i. ***VPPM_Modulator_one***

Description:

This block relates the incoming floating point stream (0s and 1s representing bits) at a transmitter to the transmitted two-level VPPM modulated signal for the given values of maximum magnitude of the signal, minimum magnitude of the signal, number of output samples per symbol and number of high samples per pulse (duty cycle).

Parameters:

i. **Max_magnitude** – The maximum value of the modulated signal or the magnitude of the higher level of the modulated signal given as a floating point.

ii. **Min_magnitude** – The minimum value of the modulated signal or the magnitude of the lower level of the modulated signal given as a floating point.

iii. **Samples_per_symbol** – The number of output samples per each input symbol given as an integer.

iv. **Samples_per_pulse** – The number of samples per each pulse (when the signal is at the higher level) given as an integer.

ii. ***VPPM_Modulator_two***

Description:

This block relates the incoming floating point stream (0s and 1s representing bits) at a transmitter to the transmitted two-level VPPM modulated signal for the given values of amplitude of the signal, mean of the signal, number of output samples per symbol and number of high samples per pulse (duty cycle).

Parameters:

i. **Amplitude –** The value of the amplitude of the modulated signal.

ii. **Mean –** The mean value of the modulated signal.

iii. **Samples_per_symbol –** The number of output samples per each input symbol given as an integer.

iv. **Samples_per_pulse –** The number of samples per each pulse (when the signal is at the higher level) given as an integer.

5. **OOK Demodulator:**

In OOK Demodulation, the received OOK signal is converted back into data bits where each bit specify the On and Off states. Hence, only a single bit is demodulated per symbol where the higher magnitude denote the bit 1 whereas the lower magnitude denote the bit 0.

Inputs and Outputs:

The input to the OOK demodulator block is a continuous two-level modulated electrical signal while the output is a stream of bits.

Description:

This block relates the incoming real-valued two-level signal/waveform at a receiver to the floating point stream (0s and 1s representing bits) for the given values of threshold and number of input samples per symbol.

Parameters:

i. **Threshold –** The value of the modulated signal above which is considered the higher level given as a floating point. The signal above the threshold represent bit 1 while the signal below it represent bit 0.

ii. **Samples_per_symbol –**The number of input samples per each output bit given as an integer.

6. **PAM Demodulator:**

In PAM demodulation, the multi-level signal representing multiple bits per symbol is demodulated to get the original stream of data bits.

The PAM Demodulator is realized by two blocks. The first one is the actual demodulator which maps the incoming signal level to the corresponding decimal values and the second one is a bit mapper which converts these decimal values into bits.

Inputs and Outputs:

The input to the PAM demodulator block (demodulator + bit mapper) is a continuous multi-level modulated electrical signal while the output is a stream of floating point values (0s and 1s representing bits).

i. ***PAM_Demodulator***

Description:

This block relates the incoming real-valued multi-level modulated PAM signal at a receiver to the corresponding decimal value of the symbol for the given values of modulation order, maximum magnitude of the signal, minimum magnitude of the signal and number of input samples per symbol.

Parameters:

i. **Modulation_order –** The number of symbols in the constellation corresponding to the PAM signal levels to be given as an integer value.

ii. **Max_magnitude –** The maximum value of the modulated signal or the magnitude of the higher level of the modulated signal given as a floating point.

iii. **Min_magnitude –** The minimum value of the modulated signal or the magnitude of the lower level of the modulated signal given as a floating point.

iv. **Samples_per_symbol –** The number of input samples per each output symbol given as an integer.

ii. ***decimal_to_binary_mapper***

Description:

This block converts the incoming decimal values to floating point stream (0s and 1s representing bits) for the given value of PAM signal levels (for deciding the mapping from a single decimal value to multiple bits).

Parameters:

i. **Modulation_order –** The number of symbols in the constellation corresponding to the PAM signal levels to be given as an integer value.

7. **VPPM Demodulator:**

In VPPM demodulation, the received VPPM signal is converted back into data bits where each bit specify the On and Off states.

Inputs and Outputs:

The input to the VPPM demodulator block is a continuous two-level electrical signal which is the VPPM modulated signal while the output is a stream of floating point values (0s and 1s representing bits).

Description:

This block relates the incoming real-valued two-level VPPM signal/waveform at a receiver to the floating point stream (0s and 1s representing bits) for the given values of number of output samples per symbol, number of high samples per pulse (duty cycle) and gain.

Parameters:

i. **Samples_per_symbol –**The number of input samples per each output bit given as an integer.

ii. **Samples_per_pulse –** The number of samples per each pulse (when the signal is at the higher level) given as an integer.

iii. **Gain –** The high value used in the matching signals given as a floating point. The default value is set to 1, however, higher values can improve performance in certain conditions.