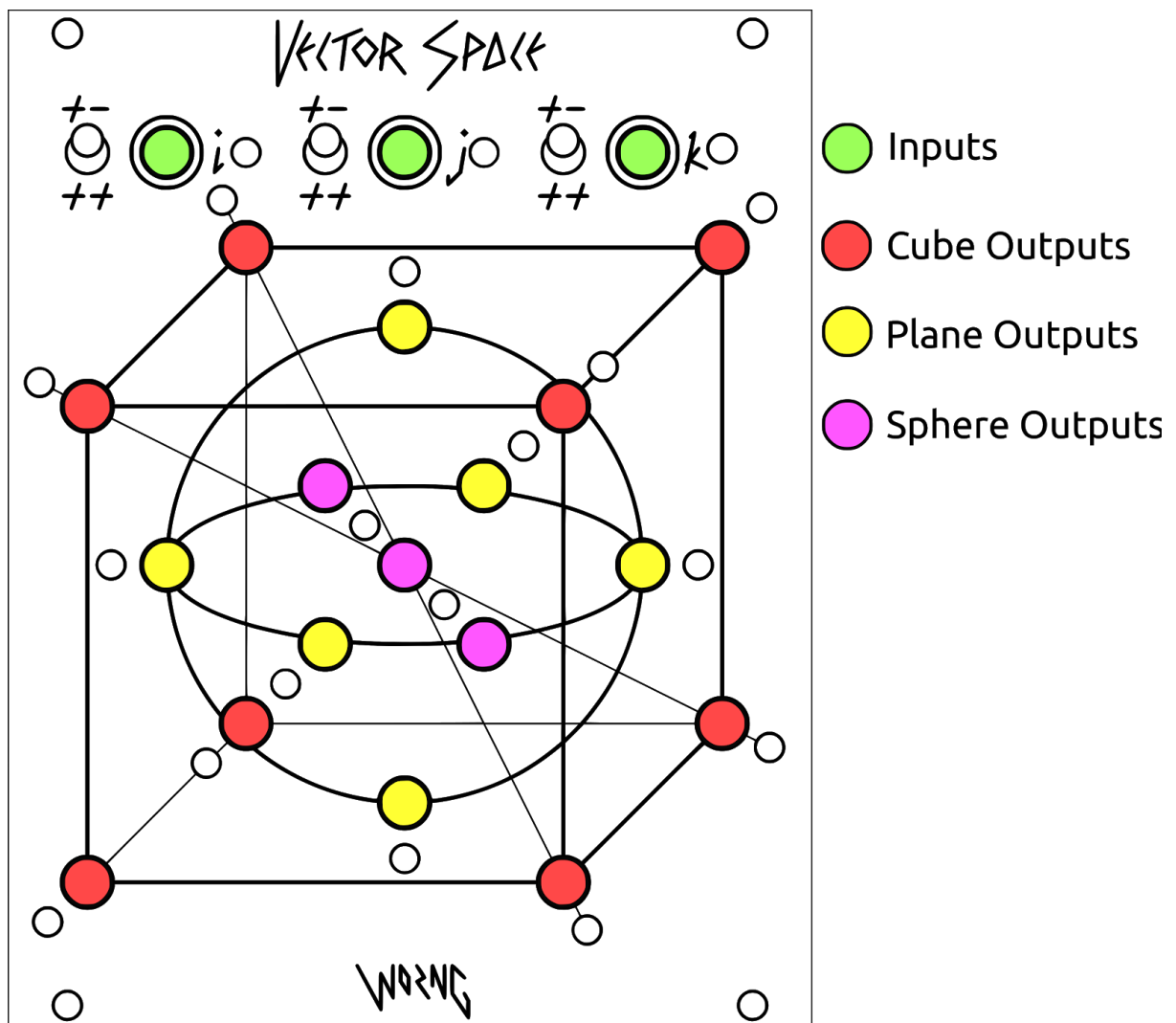


WORNG Electronics Vector Space user manual

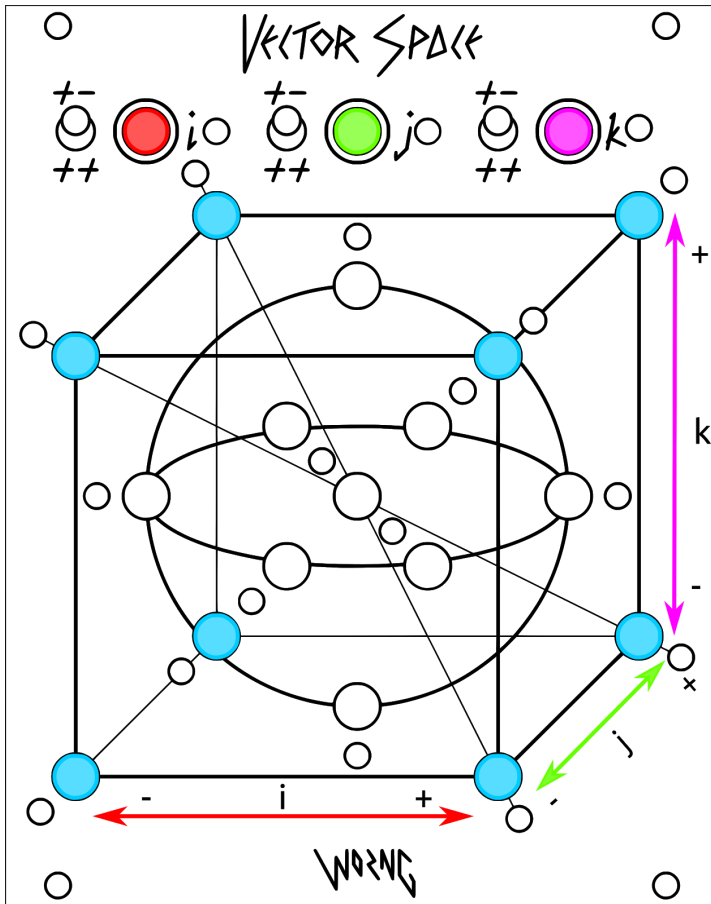
Thank you for purchasing a WORNG Electronics Vector Space.

Your Vector Space was designed as a way to increase the usability of your modulation sources by combining three sources in seventeen different combinations to give you seventeen CV outputs which are all interrelated but unique. By itself the Vector Space does very little, but combined with random voltages, envelopes, sequencers, LFOs, joysticks and more modulation sources it lets your patches get very deep, very quickly. Vector Space is 100% analogue and will process signals into and above audio rate, and up to +/-10V.

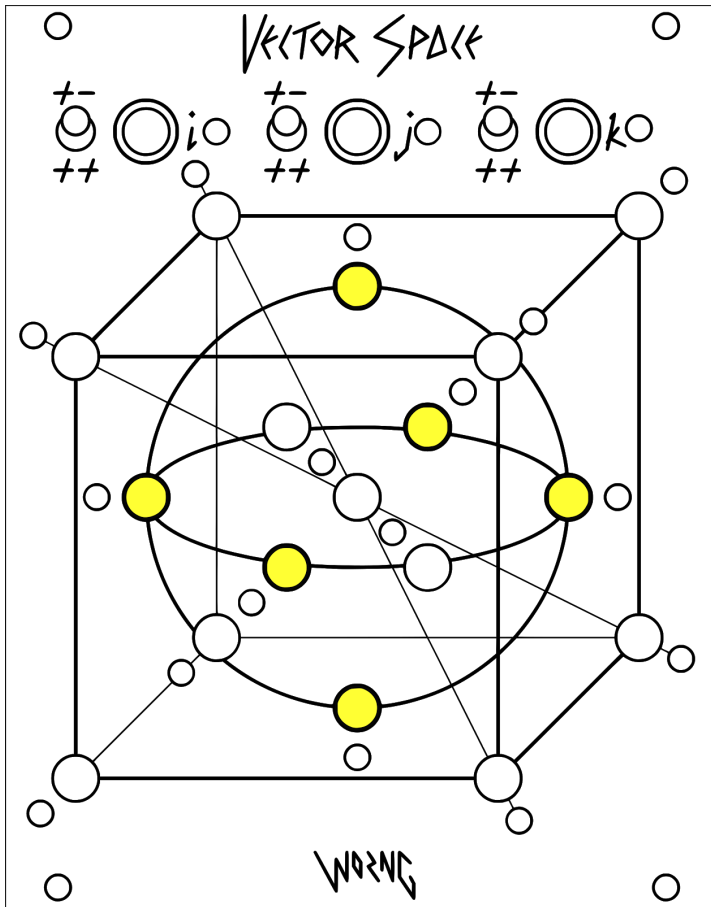


Vector Space has three inputs, **i**, **j** and **k**. A switch marked **++/+** for each input allows you to select for **+/-5V bipolar** or **0/10V unipolar** voltages. These switches can also be used as a performance control in their own right as switching them will offset the incoming voltage, whether it's unipolar or bipolar.

Vector Space has seventeen outputs, in three groups:

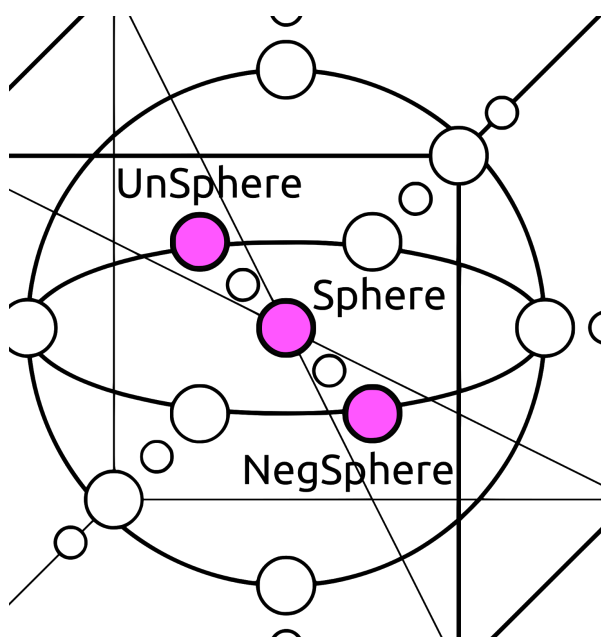


The eight **Cube** outs have bipolar Green (positive) and Red (negative) LED indicators and present the eight combinations of in and out of phase mixes of the three inputs. These outputs will give you a fairly easy to understand combination of the input sources.



The six **Plane** outs are located on the intersections of the unit sphere inside the cube and have bipolar Gold (positive) and Red (negative) LED indicators. These outputs are a combination of the rectified versions of two of the inputs summed with an in or out of phase version of the third. As these contain rectified voltages they're skewed towards the positive and at any one point in time three of the outputs will have similar voltage outputs, but as the phases of the inputs

change which three that is will change. The plane outs are capable of doubling the frequency of a bipolar modulation source by rectifying it.



The three **Sphere** outputs are located in a diagonal line in the centre of the the module. The centre output is **Sphere** and it contains a sum of all rectified inputs, this voltage corresponds with the net distance of the point from the centre point of the cube and is indicated by the pink/purple LED to its bottom right. The **NegSphere** output is

located on the other side of this LED and outputs a negative version of the Sphere output. The final output is **UnSphere** which is located above and to the left of the Sphere output and is indicated by a white LED. This output corresponds to the distance from the point to the closest edge of the unit sphere, so will be +5V when the point is in the centre of the cube and will decrease as the point moves towards the edge in any direction.

Connecting your Vector Space

Vector Space needs to be installed in a compatible Eurorack case with power supply to function. The power cable is positioned with the red stripe oriented downward on the module and is marked with the word STRIPE. Vector Space was designed for operation on +/-12V and consumes ~150mA of current per rail during normal operation.

Getting started with Vector Space

The mental model used to help visualise the outputs is that of a unit cube in three dimensional space, with the three inputs controlling the position of a point in that space (i, j and k in vector mathematics). As the point moves closer to a corner of the cube the voltage there increases, as it moves away it decreases. Don't worry if this means nothing to you, all outputs are marked with LEDs indicating their output voltage and the Vector Space really responds well to exploring it by patching one output after another to see what sounds good. Due to the open and inviting output section you will find yourself coming up with interesting combinations which you hadn't even thought of previously.

One very rewarding patching technique is to patch some of the outputs back to modulate the sources which you're feeding the Vector Space, for example FM inputs of LFOs, VC rise or fall times of envelopes, waveshapes of wavetable LFOs etc.

Patch Suggestions

Generative counterpoint:

This patch uses the unique characteristic of the Plane outputs to always have three outputs the same to generate interesting harmonic/melodic lines.

Patch three stepped voltage sources (they can be random or sequenced, clocked together, separately or with divisions) to the three inputs of Vector Space, then take at least three of the **Plane** outputs through quantisers and run them to the inputs of your VCOs. If you choose your quantiser scales correctly the VCOs will move together and apart depending on the input signals, and generate interesting combinations of notes. Try switching the input offset switches as well to add to the effect in real time.

Ultimate LFO logic:

Patch three LFOs to the three inputs, the outputs are now a smorgasbord of modulation with simulated quadrature LFOs available at the **Cube** outputs, higher frequency versions at the **Plane** outputs and particular randomesque voltages from the **Sphere** outputs.

Quad/cubic spatialisation:

Patch the X and Y of a joystick to the i (left/right) and j (front/back) inputs of the Vector Space, then take the outputs of the lower four **Cube** outs to four VCAs, and run these to four channels of your quadraphonic array. The joystick now controls the spatialisation of the sources plugged into the VCA inputs. For cubic spatialisation, add a third modulation source to the k (up/down) input and connect the upper four **Cube** outs to four more VCAs and the upper speakers of your cubic array.

3D Vector Synthesis:

This patch is a riff on the Vector Synthesis of the Sequential Circuits Prophet VS. Patch five waveform outputs of a VCO to five VCAs, and then the VCAs to a mixer. Patch a joystick or another pair of modulation sources to the i and j inputs of Vector Space, then take the lower four **Cube** outs and the **Unisphere** output and patch them to the inputs of the five VCAs. Try a fairly plain waveform like sine or triangle to the Unisphere VCA, then it can be the default timbre in the middle of your patch with other more harmonically rich timbres in the corners. Modulating the inputs to Vector Space will now scan through the waveforms, allowing you to morph between the timbres produced by the oscillator. Adding a third modulation source allows you to use the upper **Cube** outs for another four VCAs, doubling the amount of timbres available to scan through, for example you could take four waveforms from a second oscillator and mix them to another mixer, and then pan your two mixes left and right.

Timbral Garden:

Patch three oscillator outputs to the i, j and k inputs of Vector Space. They can be from three different oscillators or a single one, but the results will be more pronounced when the signal inputs are quite close in frequency. Now explore the garden of different timbres presented on the seventeen outputs of Vector Space by patching them into your audio path. The different phase relationships of the input signals will lead to frequencies being attenuated or accentuated depending on which outputs you're monitoring which is why using signals of a similar frequency is advised. The **Cube** outputs will give fairly straight variations on the original timbres, the **Plane** outputs will have higher frequency partials added to the signal. The **Sphere** outputs will tend to have a fairly large DC offset on them and will be less useful in this patch, but try them anyway and see what you get. The open layout of Vector Space allows you to easily audition all seventeen combinations to find the timbre that works for your patch. Take multiple timbres and pan them around for super wide stereo sounds.

These patches are just a starting point for you, Vector Space contains limitless possibilities and no doubt you'll come up with some unique patches of your own while using it. We always love to hear what users have come up with using our modules so if you've done something cool with your Vector Space let us know on Facebook, Instagram or Twitter.

Go forth and patch!

WORNG Electronics