OmniPreSense

AN-017 Simple Counting Algorithm

OmniPreSense has released a simple counting algorithm for its radar sensors. The algorithm makes use of the speed, range, and/or signal magnitude filter settings to count how many objects passing by the sensor meet the set thresholds. The simple counter is not robust enough to count objects which are very close together and look like a single object to the sensor but in general can give a good approximation of the count. If the flow of the objects is controlled with set gaps between them, the count can approach 100% accurate. Future enhancements to the algorithm will compensate for objects which are close together such as two people walking side by side or a car trailing another car closely.

Counting Algorithm Operation

The simple counting algorithm is available starting with v1.3.5 firmware for the OPS241-A and OPS242-A and the v1.0.0 firmware for the OPS243-A. The counting algorithm is off by default and can be turned on with the IG command. The algorithm will look at the data reports over time and any set of reports which meet the thresholds set will trigger a count. The count will continue to accumulate over time and can be read out with the N? command. The count can be cleared at any time with the N! command.

The filter settings are the same filter settings previously available for setting filters on speed (-A Doppler sensors), range (-B FMCW sensors), and signal magnitude. Figure 1 shows the filtering options while Table 1 lists the default filter settings.

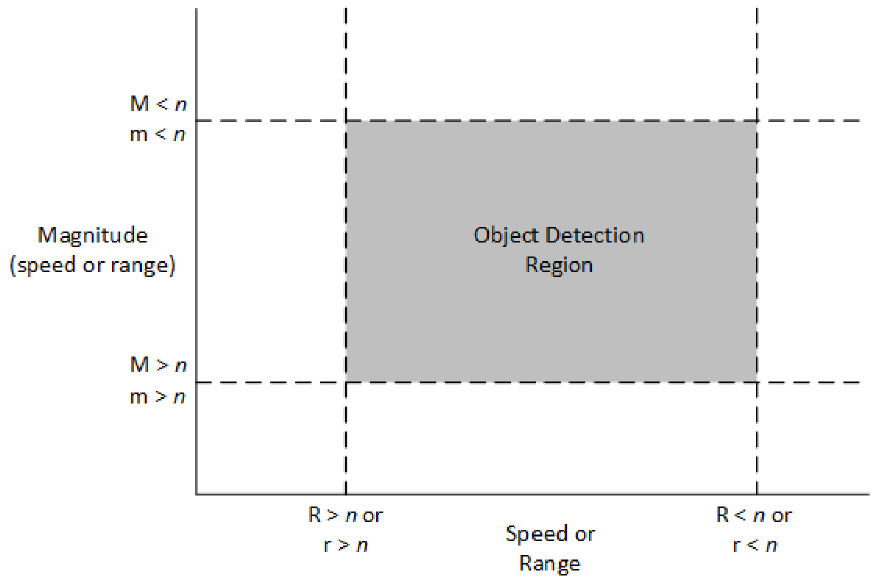


Figure 1. Speed, Range, and Magnitude Filtering

Table 1. Default Filter Thresholds

Parameter	Sensor Type	Default Value	
		Min	Max
Speed	Doppler	0	None
Range	FMCW	0	None
Signal Magnitude	Doppler	10	None
Signal Magnitude	FMCW	150	None

The algorithm works by tracking reports that exceed the filter thresholds. When two consecutive reports exceed the threshold, a start count event is been triggered. The algorithm will watch for reports that continue to meet the threshold limits and only after four consecutive missed reports will it increment the object counter. The values to start a count (default two) and increment the count (default four) is programmable. Use the N>*n* to set the start count threshold and the N<*n* to set the count end threshold.

An example of a stream of reports is show in Figure 2. In this stream of data, no speed reports are seen above the speed threshold, 0.20 m/s for the first seven reports. On the eight report a value is seen in meeting the speed threshold. The ninth report likewise meets the threshold and therefore sets the algorithm to start the count.

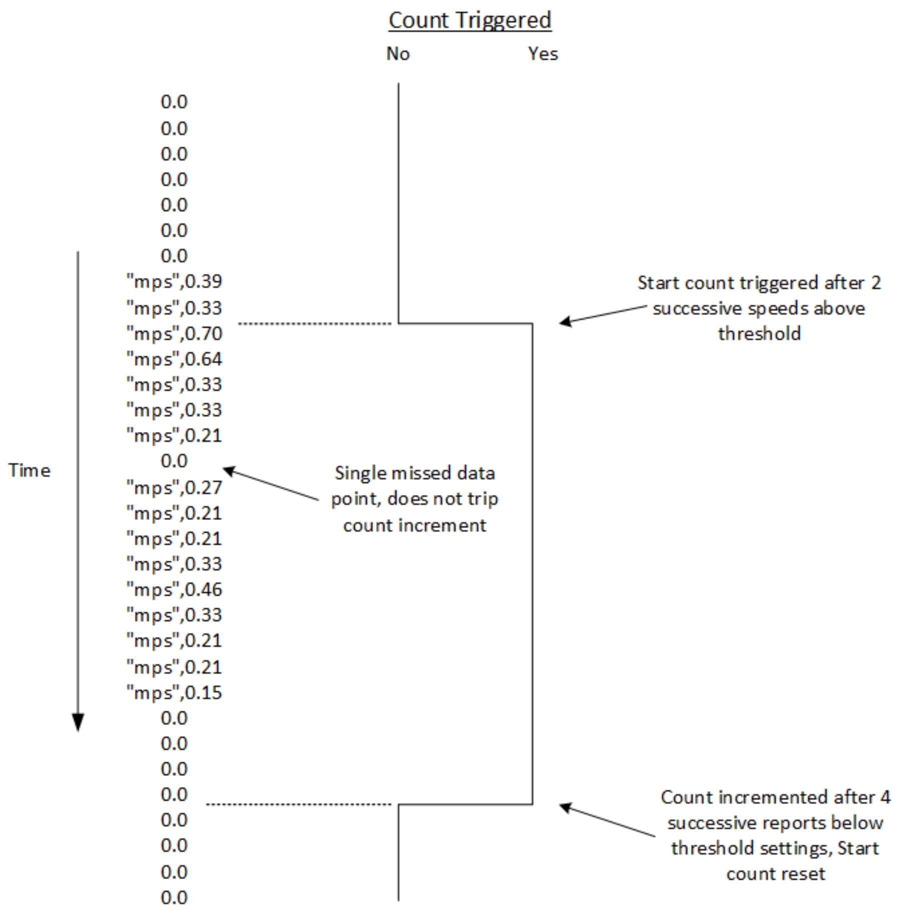


Figure 2. Simple Count Algorithm Example

The reports continue to meet the speed threshold with one exception in the middle of the stream. Since this was only a single miss, it does not trigger the count increment. After the 0.15 m/s report, four consecutive misses are seen. On the fourth miss, the counter is incremented, and the start count is reset to no count being tracked.

Counting Scenarios

The simple counter can be used for counting many objects, from people to cars to drones in the sky. If the objects are spaced apart and not close together in time and space, the simple counter should provide an accurate count. However, if the scenario is such that objects can be close together and will be seen as one by the radar sensor, then they'll only be counted as one.

An example of this is two people walking side by side. The radar sensor will mostly likely not be able to discern the differences between the two people and count them as one. Likewise, a stream of cars that are close together could be seen as a single long car should the gaps in reports never exceed the four reports threshold.

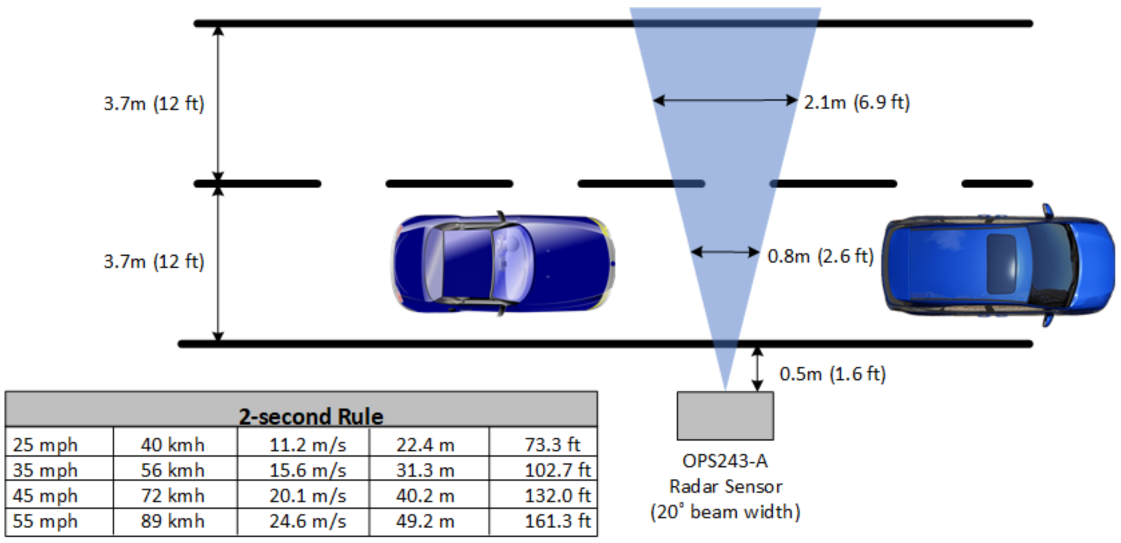
There are a few adjustments that can be made to help improve the counting algorithm and chance for a more accurate count. These include adjusting the start and stop count threshold values as mentioned earlier, using a faster report rate so more reports are seen between objects, or adjusting the position of the sensor so the traffic flow is confined to a set stream or so the sensor can more easily see individual objects.

Using a faster sample rate can help provide more reports per second and effectively chops up the distance finer between the objects. This is more effective with vehicles where they tend to follow each other. On the Doppler radar sensors (-A models), the default sample rate is 10ksps. This results in a report approximately every 155ms (slightly faster with the OPS243). By moving to a 20ksps (S2 command), the reports are every 105ms or 32% faster. At a 50ksps (SL command), reports come every 73ms.

Setting up a proper position for the sensor can also help significantly. As an example, setting the OPS243-A with its 20˚ beam width, to look perpendicular into a two-lane road provides a fairly narrow space of observation between cars. This is shown in Figure 3. A typical rule of thumb for a safe distance between cars is 2-3 seconds. Using 2 seconds, Figure 3 shows that the narrow beam width provided by the OPS243-A results in a car separation distance of between 0.8 to 2.1m while the shortest safe distance is 22m at the equivalent of 25 mph/40 kmh. In these cases, the count algorithm should provide an accurate count.

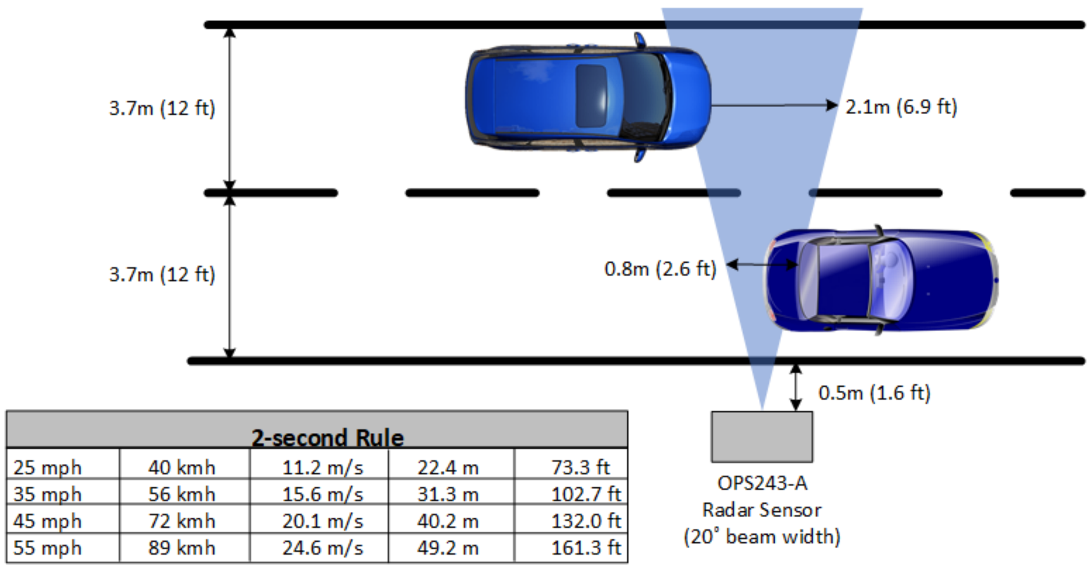
However, there will be cases where the count can provide an incorrect number. Should a car in lane 1 be trailed by another car in lane 2 (Figure 4), there is chance they would be counted as a single car. The gap between the cars would need to be less than 2m to run into just such a case.

Over time OmniPreSense will enhance the counting method to compensate those these corner case scenarios and provide for improved accuracy.



2-second Rule				
25 mph	40 kmh	11.2 m/s	22.4 m	73.3 ft
35 mph	56 kmh	15.6 m/s	31.3 m	102.7 ft
45 mph	72 kmh	20.1 m/s	40.2 m	132.0 ft
55 mph	89 kmh	24.6 m/s	49.2 m	161.3 ft

Figure 3. Optimal Counting Sensor Position



2-second Rule				
25 mph	40 kmh	11.2 m/s	22.4 m	73.3 ft
35 mph	56 kmh	15.6 m/s	31.3 m	102.7 ft
45 mph	72 kmh	20.1 m/s	40.2 m	132.0 ft
55 mph	89 kmh	24.6 m/s	49.2 m	161.3 ft

Figure 4. Miscount Count Scenario

Appendix

Command	Name	R/W	Value
N?	Query Count	Read	Reports number of objects counted. {"DetectedObjectCount":3}
N!	Reset Count	Write	Resets the number of objects in counter. {"DetectedObjectCount":0}
N>*n*	Count Start Threshold	Write	{"MotionSignal":"Status", "CountToPass":2, "CountToFail":4}
N<*n*	Count End Threshold	Write	{"MotionSignal":"Status", "CountToPass":2, "CountToFail":3}

Revision History

Version	Date	Description
A	May 22, 2019	Initial release.