Tutorial: Rounding Affects Overflows in the Shoulder Zone

Andy Bartlett MathWorks 1/30/2020

Rounding of numeric results is a necessity in both scientific and engineering calculations if you want those calculations to be fast. It is widely understood that the choice of rounding method impacts the size of round off error. A less known fact is that rounding can also affect whether quantization causes an overflow. Rounding choice affects overflow in a small portion of the input domain that this article calls the "Shoulder Zone." This tutorial will briefly explain the effect of rounding choice on inputs from the Shoulder Zone.

1. Contents

2. Example

Let's jump right to an example where the output type is a signed 3-bit integer with representable values

-4, -3, -2, -1, 0, 1, 2, 3

Figure 1 shows the result of converting input values to this type using rounding toward zero and handling overflows with saturation. The figure shows red, green, and yellow zones that relate to the question, "Did and overflow happen?" For the red and green zones, the answer is certain and is independent of rounding mode and value. For the yellow Shoulder Zone, the answer depends on the rounding mode and in some case the input value.

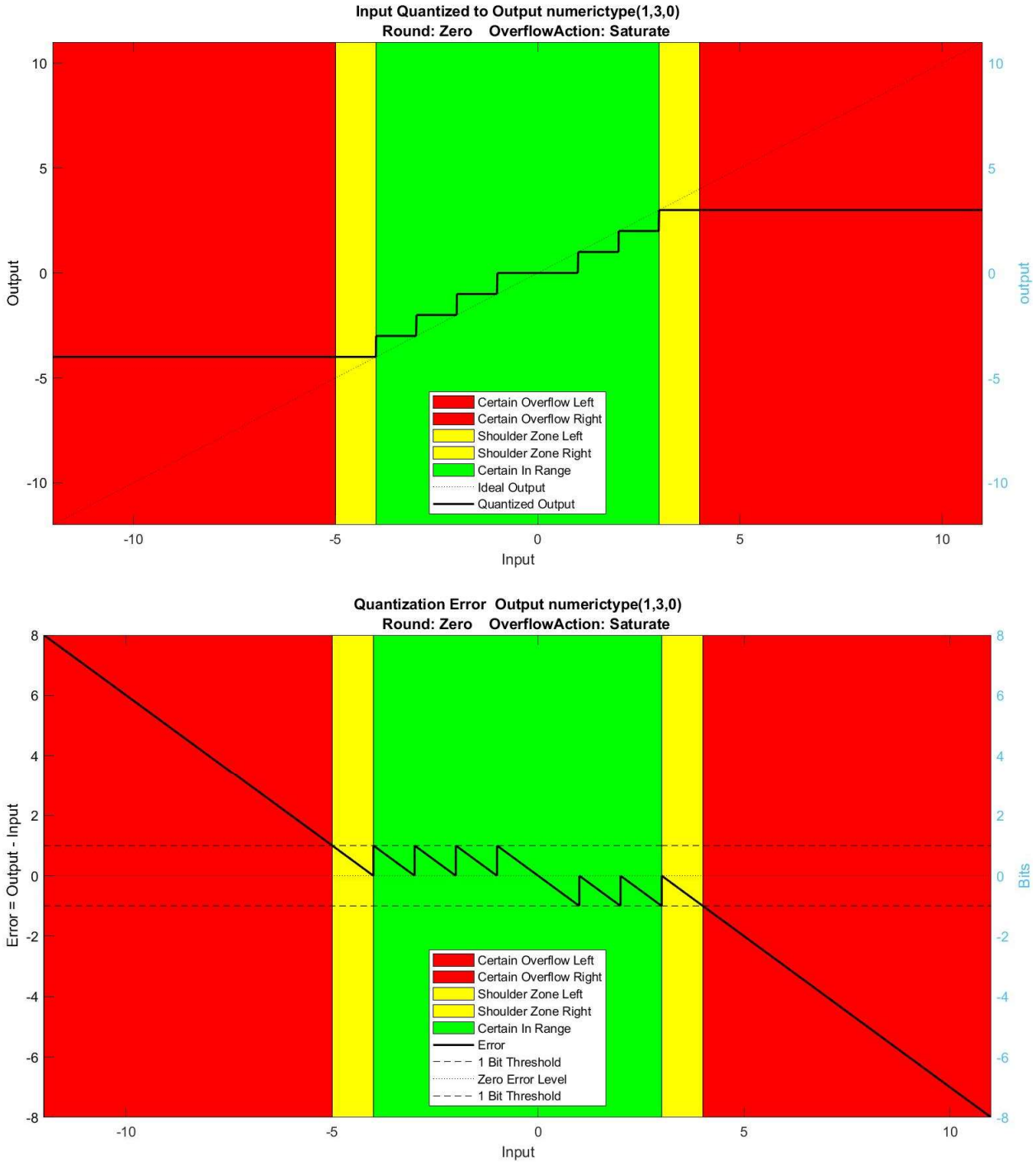


Figure 1: Quantized Output vs Input. Rounding is toward zero. Overflows are handled with saturation.

3. Red Certain Overflow Zone

The red zone represents input so far below output minimum representable value, -4, or so far above the maximum output representable value, 3, that overflow will occur with certainty. In the red zone, the

quantization error is always greater than or equal to 1 bit and can be very big. In the red zone, the occurrence of overflow does not depend on rounding mode and is certain. For simulations in Simulink, signal diagnostics would always classify red zone cases as overflow.

4. Green In-Range Zone with Certain "Mere" Rounding

The green zone represents values in the representable range from -4 to 3, inclusive. In the green zone, the quantization will be a "mere" rounding error that is always less than 1 bit. In the green zone, the limitation of the error to "mere" rounding does not depend on rounding mode and is certain. For simulations in Simulink, signal diagnostics would never classify green zone cases as an overflow.

5. Yellow Shoulder Zone

The third yellow region is called the "Shoulder Zone." Depending on the rounding mode, input values in the Shoulder Zone might involve an overflow or a "mere" precision loss. For simulations in Simulink, whether signal diagnostics classify a case as an overflow depends on the rounding mode and for nearest rounding on the value too.

The Shoulder Zone is the open interval one bit wide (aka one eps wide) to the left of the minimum finite output representable value and to the right of the maximum finite output representable value, respectively.

For the example, if the output type is a signed 3-bit integer, the two Shoulder Zones are

Left Shoulder Zone	$-5 < u < -4$
Right Shoulder Zone	$3 < u < 4$

6. Understanding Shoulder Zone Behavior: Think Rounding First

The difficulty understanding Shoulder Zone behavior can be eliminated if you understand that, by design, rounding happens first. Only after rounding is done is the question of overflow handled. By design, the classification and handling of signal overflows only consider the rounded value, not the original value.

Conceptually, the type being rounded to has the same spacing (aka precision) as the final output type but has a wider range of representable values. For the example where the final output type is a signed 3-bit integer, the type to round to has representable values

... -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, ...

So, a Shoulder Zone value like -4.3 must either be rounded down to **-5** or up to **-4** depending on rounding mode. If the rounded value is **-5** which is outside the representable range of the final output data type, then the conversion is classified as an overflow. If the rounded value is **-4** which is one of the representable values of the output data type, then the conversion is not classified as an overflow.

Note, if you use Embedded Coder or HDL Coder to generate an implementation of a data type conversion, you may not see a clear separation of rounding first, then overflow handling second. This is

a good thing and not an error. The behavior of the generated code will agree with the defined behavior. But, the operations may be fused or rearranged to provide the defined behavior with faster and/or smaller code.

7. Review Case: Rounding Toward Zero

Figure 1 shows the case of rounding toward zero. Notice that the yellow Shoulder Zone would look completely natural if it were included as part of the green zone. This agrees with the round first thinking. In the left Shoulder Zone (-5 < u < -4), the inputs will all be rounded up to -4 which is an in-range value. Likewise, in the right Shoulder Zone (3 < u < 4), the inputs will all be rounded down to 3 which is also in-range. This explains why, for zero rounding, both Shoulder Zones behave like in-range cases.

8. What Changes with Wrapping Overflows

Figure 3 shows the case of rounding toward zero with overflows configured to wrap around. For all possible rounding modes, the overflow action setting never affects the green zone. For round to zero, the Shoulder Zone is free of overflows, so overflow action does not affect the yellow region in this case.

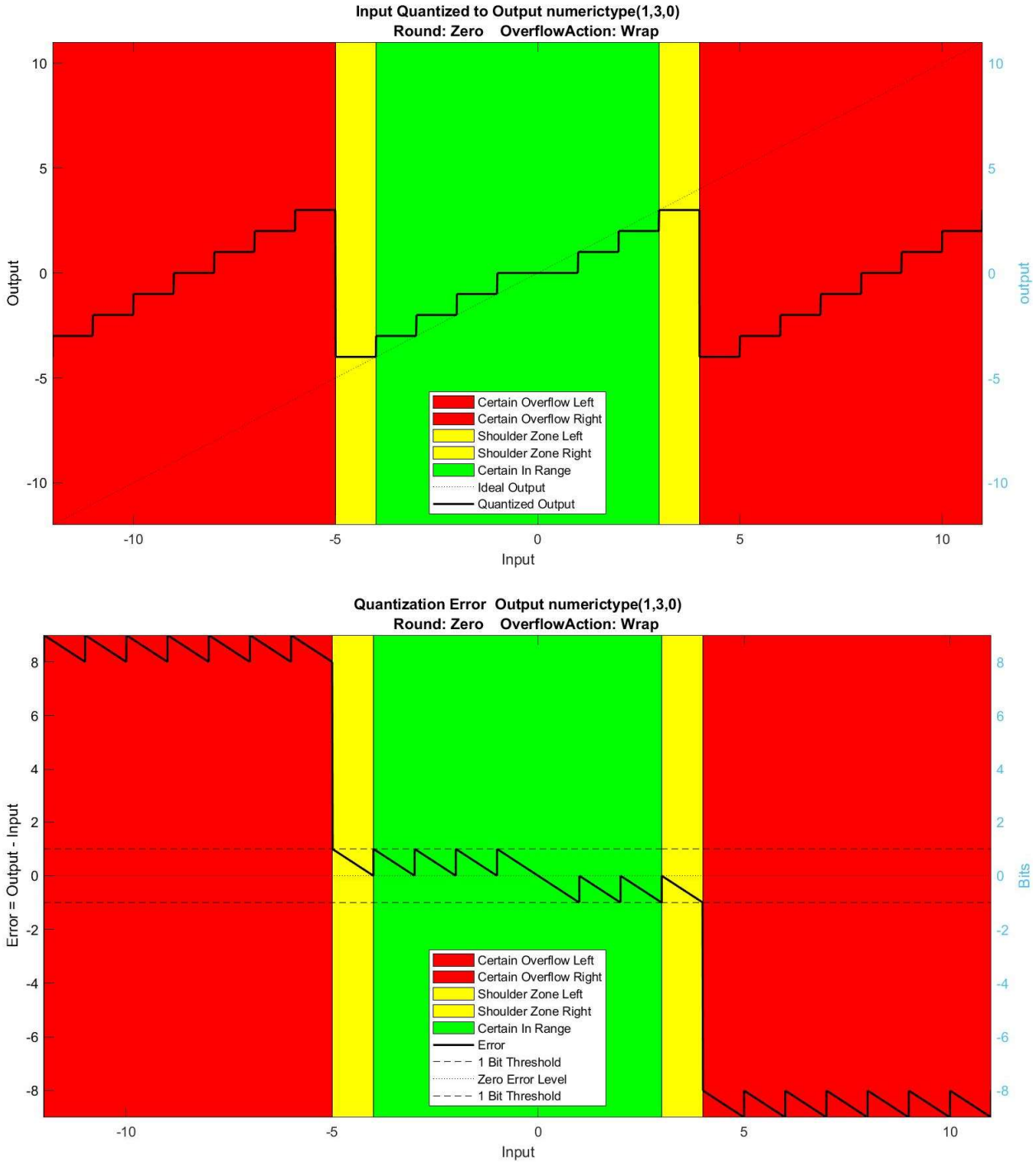
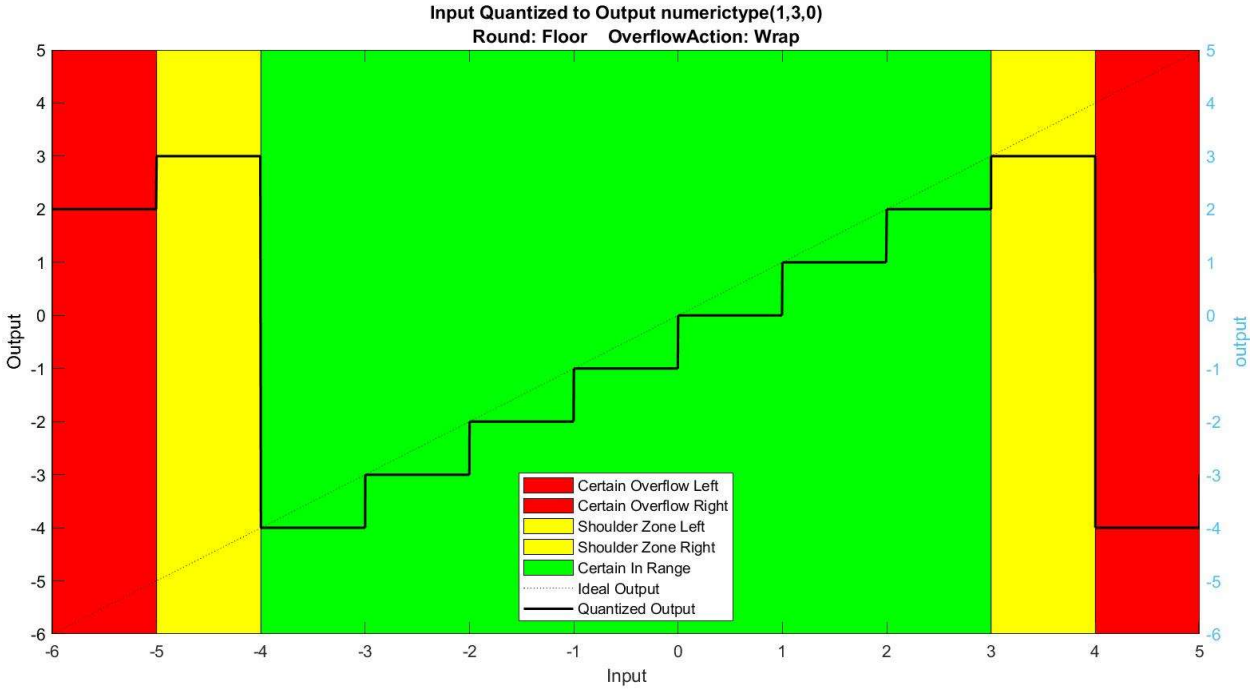


Figure 2: Quantized Output vs Input. Rounding is toward zero. Overflows wrap around modulo style.

In contrast, the red Certain Overflow Zone is dramatically impacted by the overflow action. This large impact equally affects the red zone for all possible rounding modes. For remaining examples, there is no need to discuss the red zone.

9. Review Case: Rounding Toward Floor

Figure 3 shows the case of rounding toward floor. Notice that the left Shoulder Zone looks like the red zone. This agrees with the round first thinking. In the left Shoulder Zone (-5 < u < -4), the inputs will all be rounded down to -5 which is below the minimum representable value of the output type. In contrast, the right Shoulder Zone looks like the In-Range Zone. For the right Shoulder Zone (3 < u < 4), the inputs will all be rounded down to 3 which is an in-range value. So, for round to floor, left Shoulder Zone overflows and right does not.



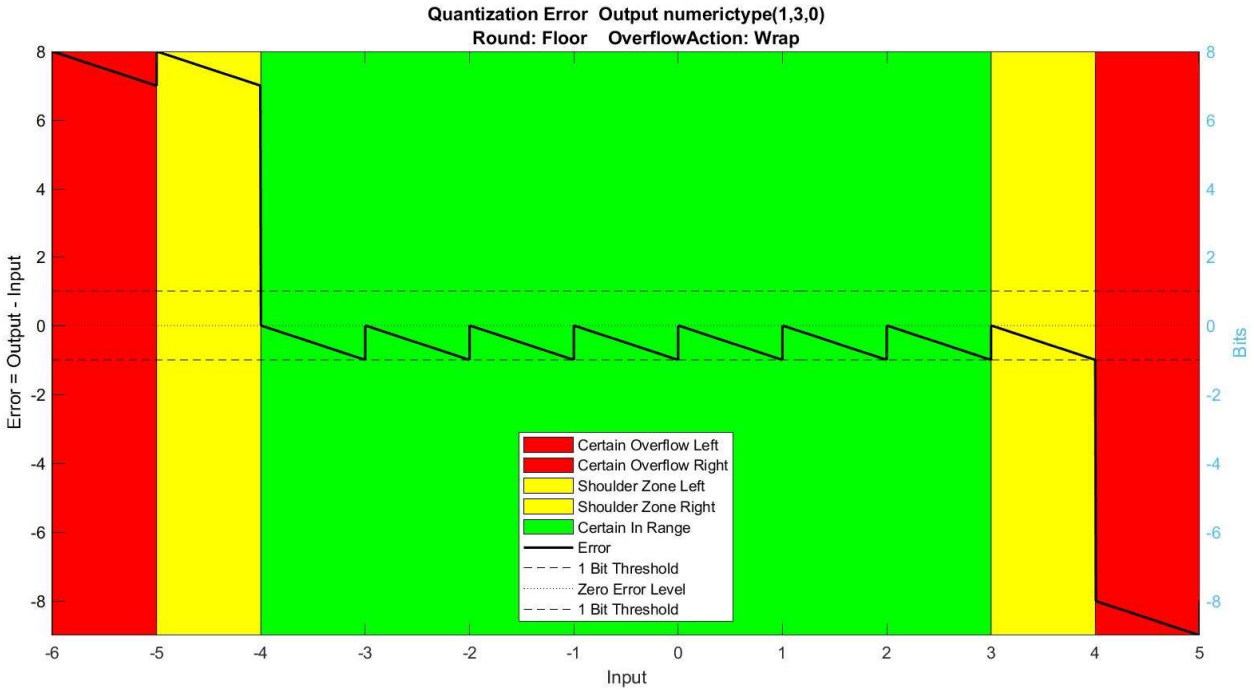


Figure 3: Quantized Output vs Input. Rounding is toward Floor. Overflows wrap around modulo style.

10. Rounding Toward Floor Dropping Only Precision Bits

For fixed-point cases, a very common form of conversion involves keep the range bits and only dropping some of the precision bits. Figure 4 shows a case where 12 precision bits have been dropped. This case only has a green In-Range Zone and the right Shoulder Zone. For round to zero, the right Shoulder Zone never has overflows. Thus, this common case has no overflows for all possible input values.

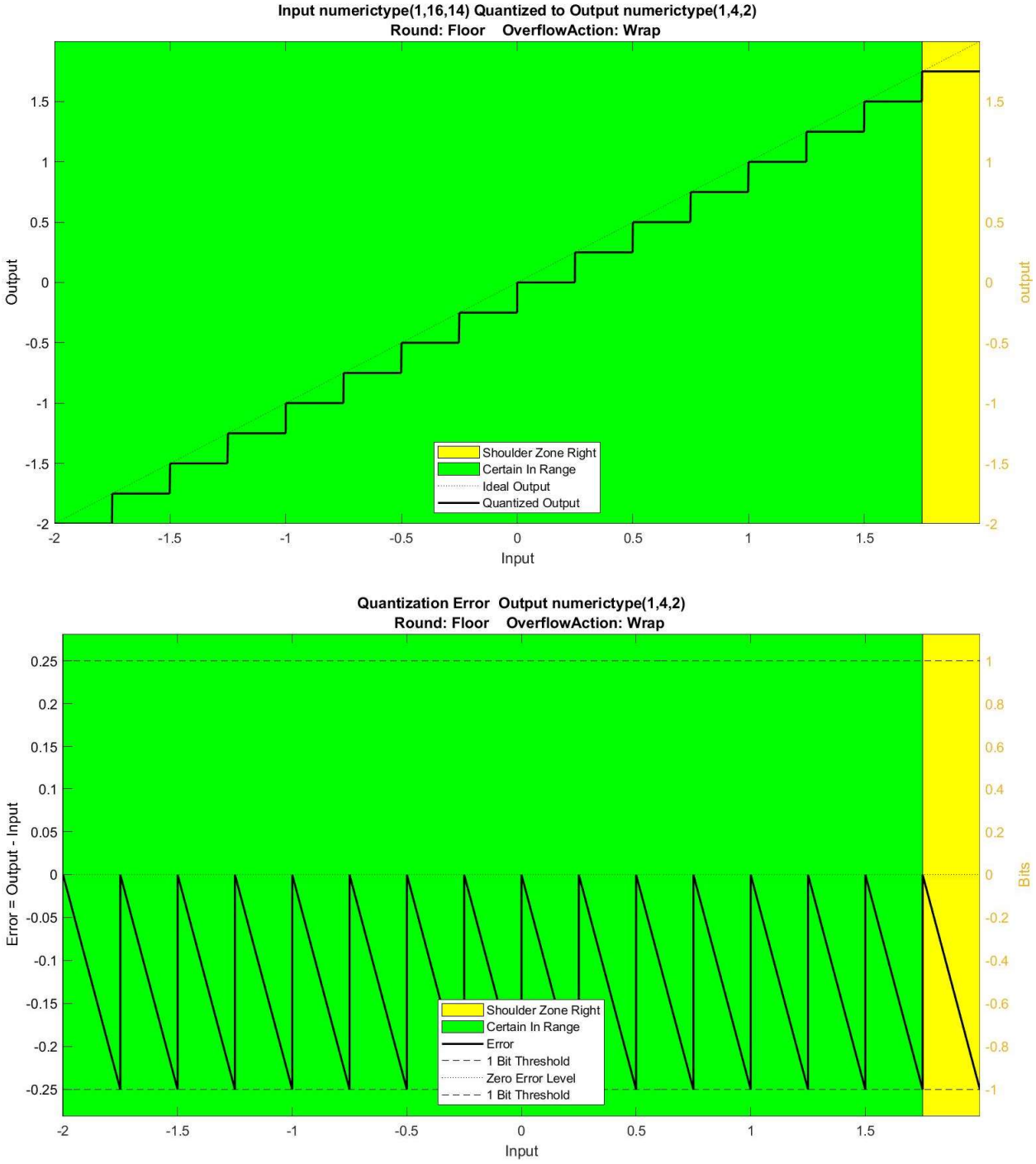


Figure 4: Quantized Output vs Input where only precision bits are dropped, and rounding is toward Floor.

11. Review Case: Rounding Toward Nearest

Figure 5 shows that round to nearest is quite different than the prior cases. Notice that both the left and right Shoulder Zones have split behavior. The outer half overflows, but the inner half does not. For the outer left Shoulder Zone (-5 < u < -4.5), the inputs all round down to -5 and thus overflow. For the inner left Shoulder Zone (-4.5 <= u < -4), the inputs all round up to -4 and are in-range. A similar analysis applies to the halves of the right Shoulder Zone.

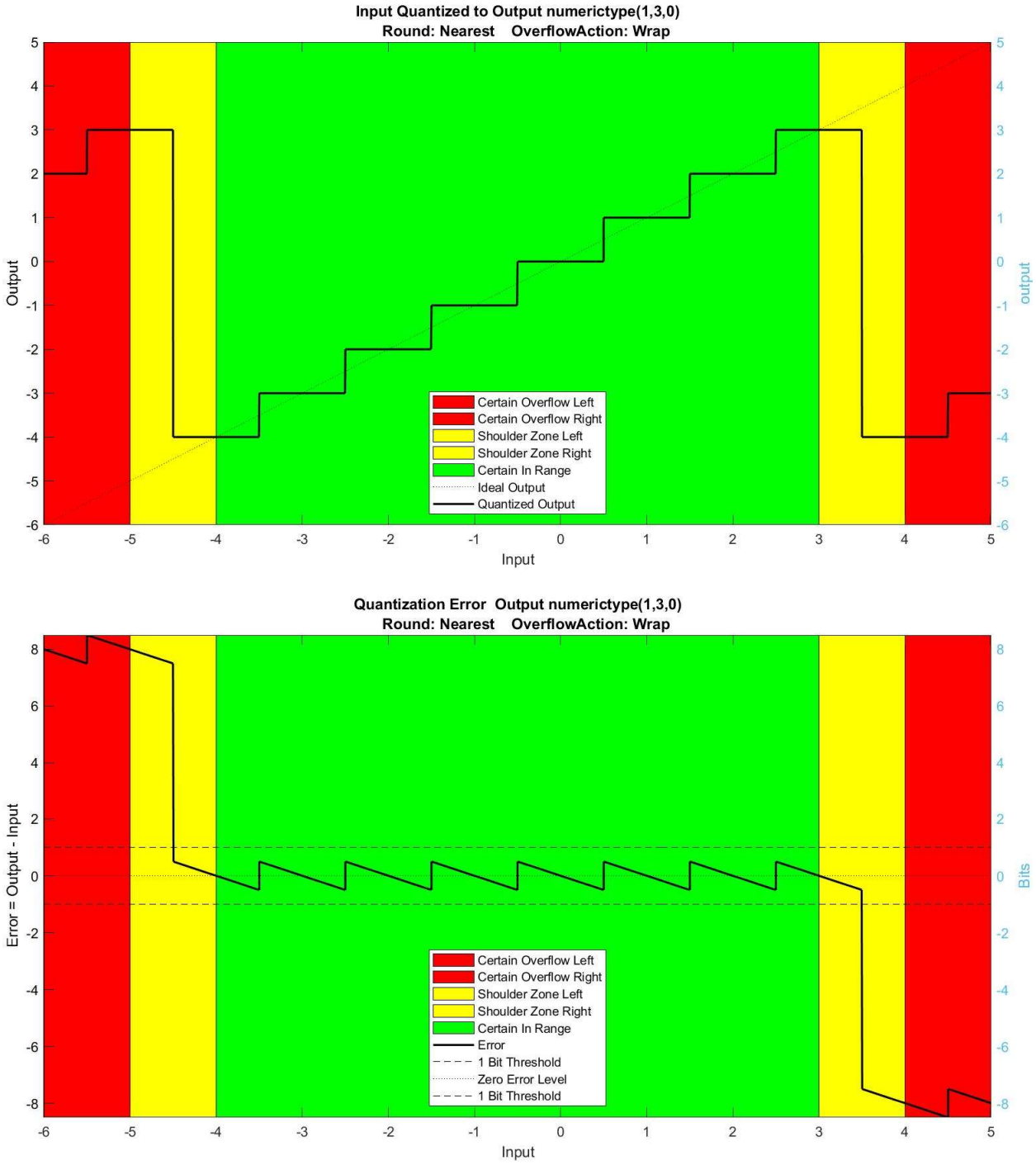


Figure 5: Quantized Output vs Input. Rounding is toward Nearest. Overflows wrap around modulo style.

12. Worst-case Shoulder Zone Error with Saturation

For any rounding mode except round to zero, values in the Shoulder Zone can lead to overflows. However, if overflows are handled with saturation, then the worst-case error in the Shoulder Zone will never exceed 1 bit. Figure 6 illustrates this error bound for the case of round to nearest. The outer

halves of the Shoulder Zone would still trigger signal overflow diagnostics, but the quantization error would still not exceed 1 bit in the Shoulder Zone.

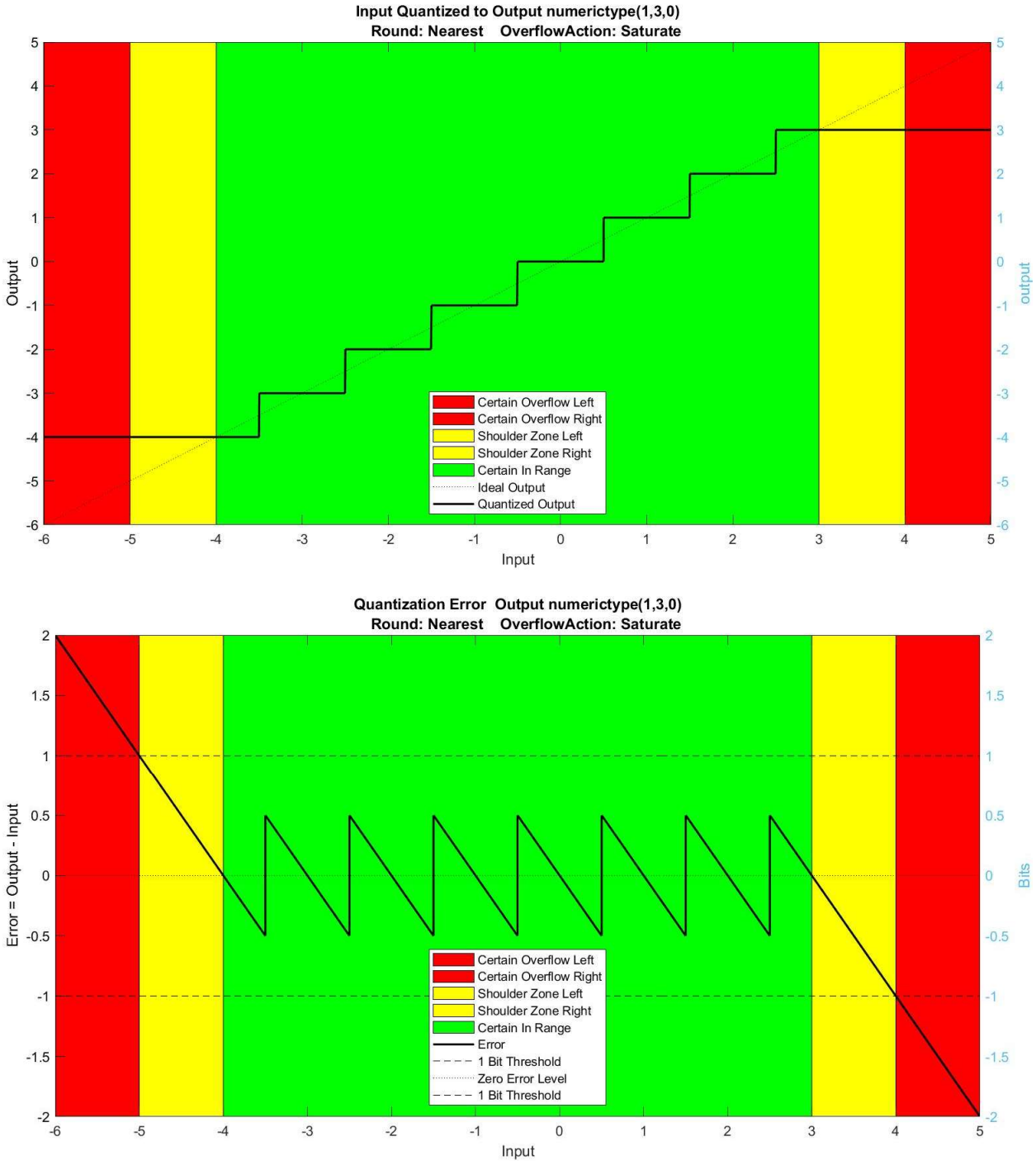


Figure 6: Quantized Output vs Input. Rounding is toward Nearest. Overflows saturate.

13. Aside: Simulink Parameter Overflow Diagnostic

Simulink classifies signal overflows as described previously in this article. Simulink classifies parameter overflows a little differently.

Simulink allows values for parameters to be entered on block dialogs using any data type, often MATLAB's default double data type. The block often uses a small data type for the parameter that will be efficient in embedded deployment of the design. When the value entered on the dialog is converted to the data type used by the block, round to nearest and saturation are always used. This is like Figure 6 where the green zone and the Shoulder Zone always have less than 1 bit of error.

For parameters, Simulink's design is to treat the following three cases as overflow free.

1. In-Range Zone
2. Entirety of both Shoulder Zones
3. The individual value on the boundary of the right Shoulder Zone and Red Zone.

Visually this is an open-closed interval starting at the left most point of the left Shoulder Zone and extending to the right most point of the right Shoulder Zone.

For Figure 3, the parameter diagnostic overflow free zone is this interval.

$$-5 < u <= 4$$

The entire parameter diagnostic overflow free zone the quantization error is always less than or equal to 1 bit.

The 3rd case, i.e. including the right most point, is designed to allow entering the extreme values of a fixed-point data type with less fuss. Consider a few example

Data Type	Min Representable Value	Max Representable Value
numerictype(1,16,13)	**-4**	**3.9998779296875**
numerictype(1,32,29)	**-4**	**3.999999998137354850769042968 75**
numerictype(1,64,61)	**-4**	**3.999999999999999995663191310057982263970188796520233154296875**

Clearly, it is painful to enter the correct maximum value for each data type. The last row is especially challenging because that value is more precise than double can handle. If you enter that text for that value in MATLAB, it will just evaluate to 4.

If the maximum is simply entered as 4. The value will be converted to the maximum shown in the table. The conversion will involve exactly 1 bit of error. Because this corresponds to case 3, the exact boundary of the right Shoulder Zone and the Red Zone, Simulink will NOT classify that parameter conversion as an overflow. The exact 1 bit of error would be covered by Simulink's Parameter Precision Loss diagnostic, but not the Parameter Overflow diagnostic.

14. Summary

This article has shown that data type conversion can be split in to three zones, **Certain In-Range Zone**, **Certain Overflow Zone**, and **Shoulder Zone**. The following facts apply.

- In-Range Zone always involves “mere” rounding errors of less than 1 bit
- Overflow Zone always involves quantization errors that are 1 bit or more, often much more.
- Depending on the rounding mode, portions of the Shoulder Zone may or may not overflow.
    - For zero rounding, no overflows occur
    - For nearest, the outer halves overflow
    - For floor, the left shoulder overflows
        - There is no left shoulder for casts that only drop precision bits.
    - For ceiling (not shown), the right shoulder overflows
- If saturation is on, then quantization errors for the Shoulder Zone never exceed 1.

This article has also discussed classification of overflows. For signals, rounding is always done first. If the rounded value is out of range, then an overflow has occurred. For parameters, the open closed interval beginning with the left Shoulder Zone and ending with the right Shoulder Zone is always classified as overflow free.

Thank you for reading

Andy Bartlett

andyb@mathworks.com