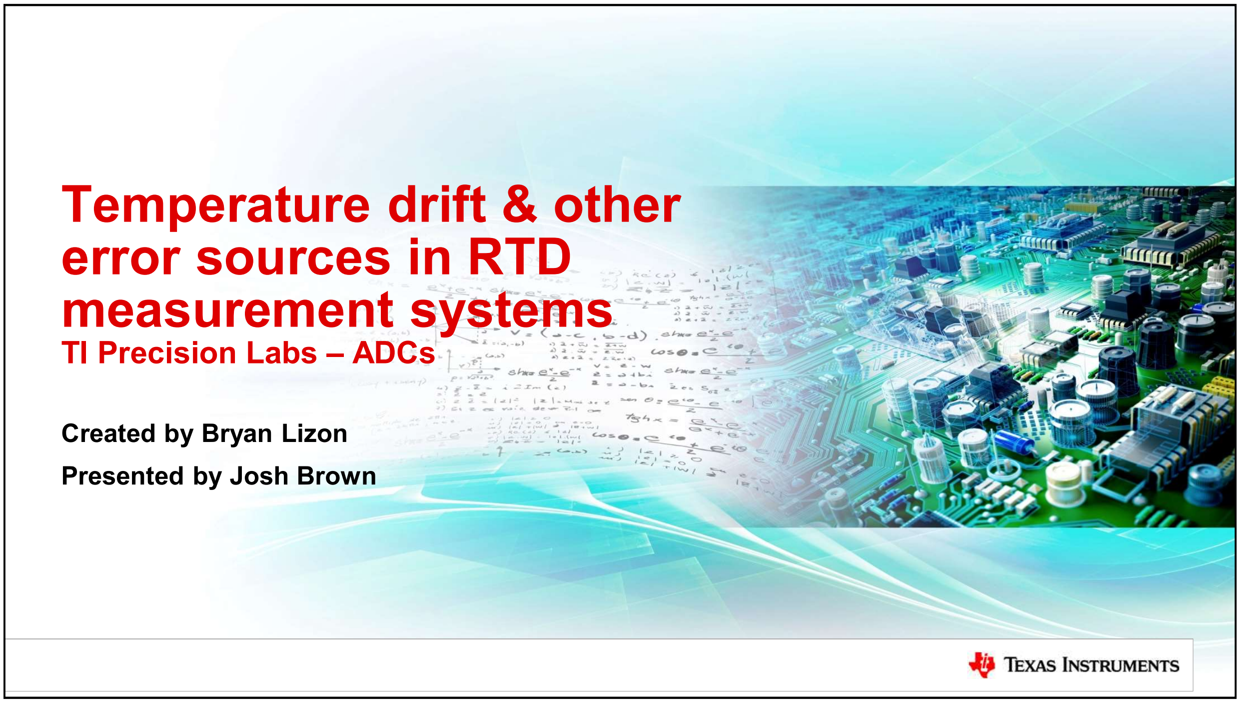
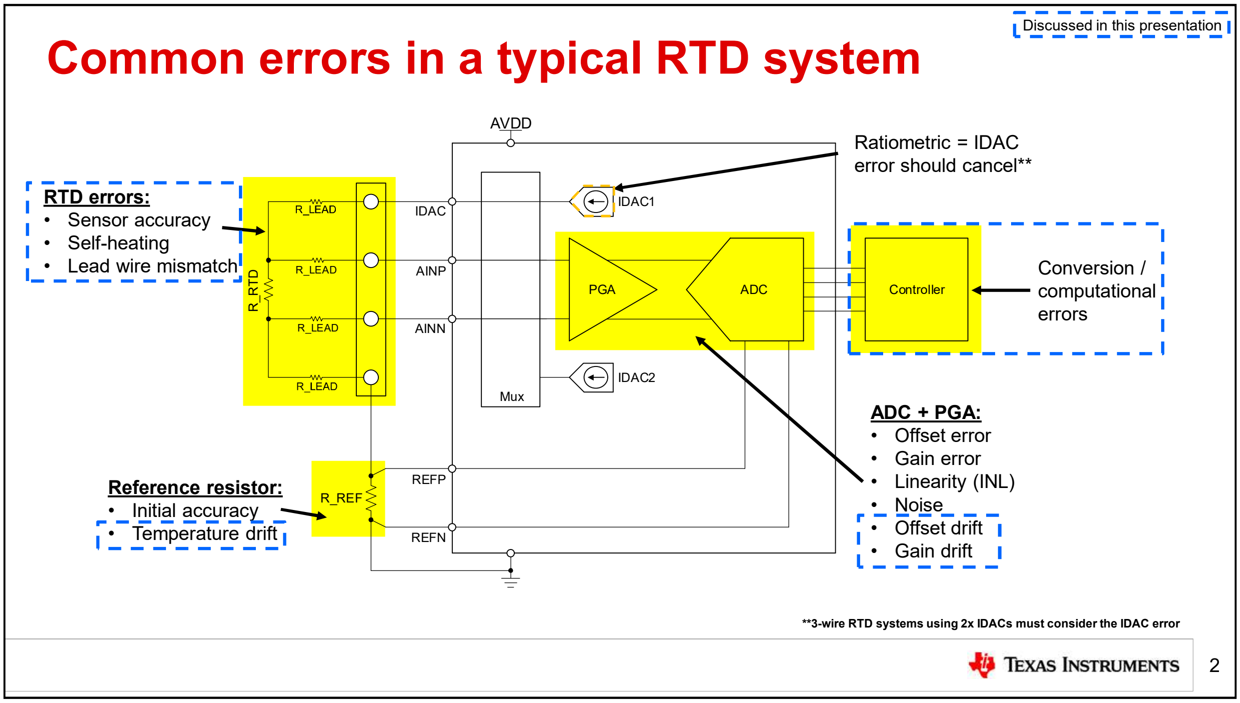


Hello, and welcome to the TI Precision Labs module introducing temperature drift and other error sources in RTD measurement systems. A previous Precision Labs module introduced all error types and provided a detailed analysis of initial errors in an RTD measurement system. This module continues that analysis to discuss how these errors are affected by a change in temperature, as well as additional system error sources

To begin, let’s review the different error sources in an RTD measurement system



Shown here is a typical RTD measurement system that includes a 4-wire RTD connected to a terminal block, a low-side reference resistor, an ADC with an integrated PGA and current sources, and a controller to perform the resistance to temperature conversion. This system has several common error sources

The ADC and amplifier can contribute multiple types of error. In this image, the amplifier is integrated into the ADC package, but the same error analysis applies for a discrete amplifier. In either case, the amplifier and ADC can contribute offset and gain error, linearity errors, and drift errors. While both components can also contribute noise, this topic is not discussed further in this presentation. Instead, refer to the Precision Labs series on ADC noise for a detailed noise analysis

The precision resistor used to establish a ratiometric reference voltage can contribute initial accuracy and temperature drift errors. We will assume there is no IDAC error because of the ratiometric reference configuration. However, IDAC mismatch error does need to be considered for 3-wire RTD measurement systems using two IDACs. This topic is covered in detail in a previous Precision Labs module discussing challenges with 3-wire RTD systems. Please review that module for more information about errors related to IDAC mismatch

The RTD itself can contribute error in the form of variation in the sensor accuracy, self-heating, and lead wire mismatch. Finally, the controller can introduce computational errors in the resistance-to-temperature conversion process

This presentation focuses on over-temperature, sensor, and computational errors, which are highlighted by the blue boxes. A previous presentation provided a detailed analysis of the remaining error sources

Let’s begin by looking at some characteristics of temperature drift

**Both initial offset and offset drift can be positive or negative

Characteristics of temperature drift

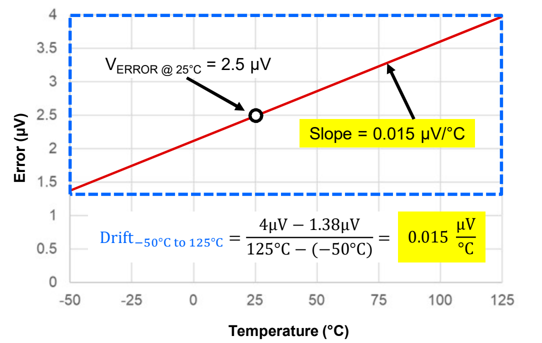
- Affects offset, gain error, linearity, and reference accuracy
- Applies to the *system* temperature, not the *measurement* temperature
- Specified using the "box" method:
  - Industry-standard for drift errors
  - Difference between the minimum and maximum errors across a given temperature range

$$\mathrm{Drift} = \frac{\mathrm{Error_{Max}} - \mathrm{Error_{Min}}}{\mathrm{T_{Max}} - \mathrm{T_{Min}}}$$

ADS124S08 offset specs (G = 8)

Condition	MIN	TYP	MAX	Unit
Offset error (T_A = 25°C)	-15	2.5	15	µV
Offset drift (T_A = -50°C to +125°C)	-0.2	0.015	0.2	µV/°C

ADS124S08 initial offset versus offset drift (G = 8)**



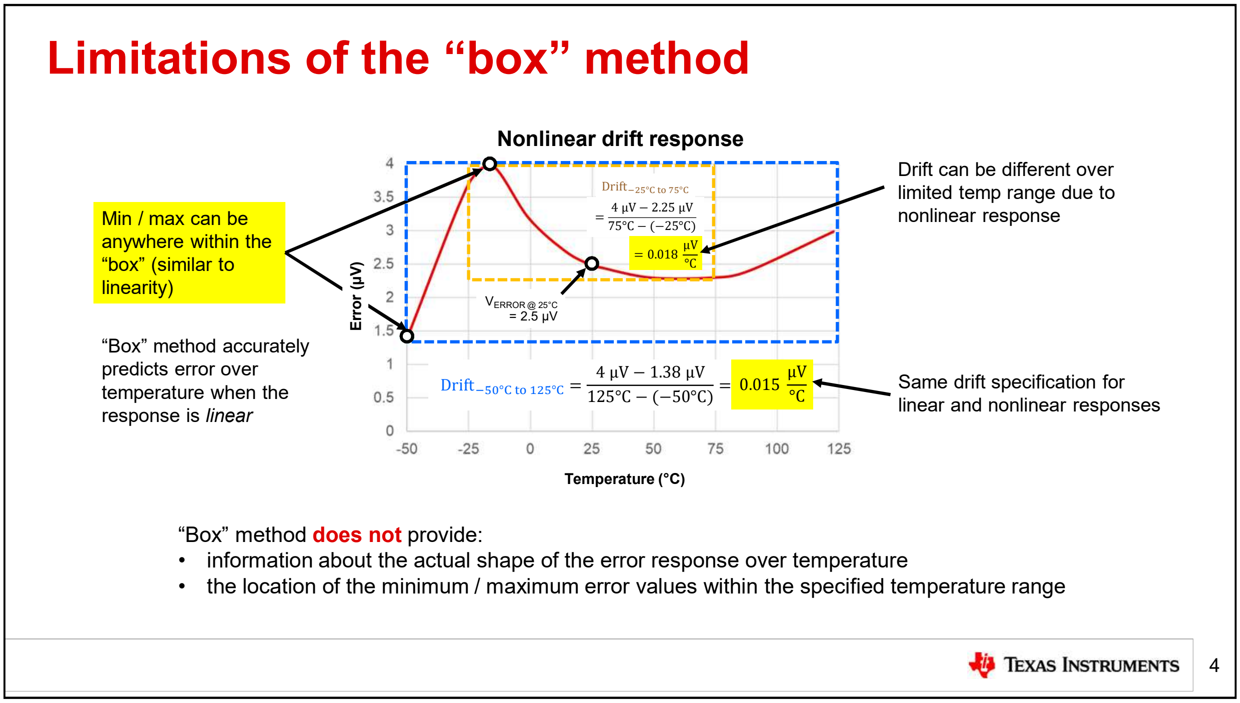
Unlike the initial errors introduced in a previous Precision Labs module, temperature drift is not a standalone error. Instead, each error source has a temperature dependency that changes how much error is introduced into the system. All of the errors discussed in this presentation are affected by a change in temperature. However, ADC datasheets typically do not give a specification for linearity drift, so we will we only focus on temperature drift errors due to offset, gain error, and the reference resistor accuracy

Another important consideration for temperature drift errors is the temperature range. Temperature drift depends on changes in the system temperature, not the measurement temperature. The system temperature applies to the signal chain including the ADC, amplifier, and reference resistor, while the measurement temperature applies to the actual RTD sensor

Temperature drift is typically specified using the box method. This industry-standard method calculates temperature drift by plotting the error across a given temperature range and finding the minimum and maximum error within this range. The equation on the left is then used to calculate the resulting temperature drift

As an example, the table and plot on the right show that the ADS124S08 initial offset error at a gain of 8 is 2.5 microvolts. Note that this is specified at 25 degrees Celsius. When we measure the ADC offset error across the entire temperature range, we might get a response similar to the red line in the plot on the right. Applying the box method in blue to this plot yields a temperature drift of 0.015 microvolts per degree Celsius across the entire temperature range. This value represents the slope of the line between opposite ends of the box, and is reported in the ADC datasheet as shown. Also note that both initial error and drift error can be positive or negative

While the box method provides some meaningful information about how an error behaves across temperature, the next slide explores some of the limitations of this method



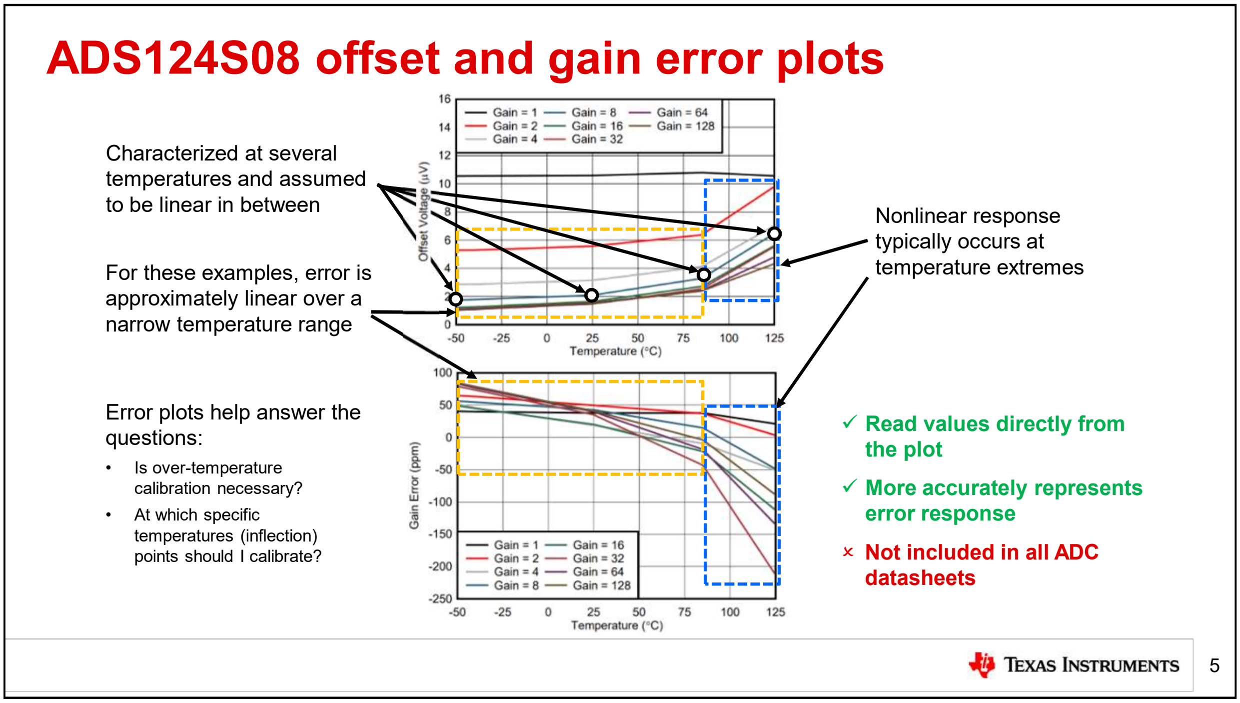
As noted on the previous slide, the box method yields a drift specification that effectively linearizes the error response. Therefore, if the error response is linear, the box method accurately predicts the error at any temperature within the range. However, what happens if the error response is nonlinear?

Shown here in red is an example of a nonlinear error response plot. Note that the initial error of 2.5 microvolts is the same as the linear response, but otherwise the shape is very different. Applying the box method in blue to this plot yields a temperature drift of 0.015 microvolts per degree Celsius across the entire temperature range, which is the same drift specification calculated for the linear error response. This similarity occurs because the minimum and maximum values can theoretically be anywhere inside the box. In other words, the worst-case error does not necessarily need to occur at either of the temperature extremes. In this example, the maximum value occurs at approximately -20 degrees Celsius

As a result, a more limited temperature range can provide different drift values depending on the actual shape of the error response. The example in orange shows that the box method yields a temperature drift of 0.018 microvolts per degree Celsius from -25 degrees Celsius to 75 degrees Celsius.

Ultimately, there are some limits to using drift values derived using the box method. First, the box method does not provide any information about the actual shape of the error response over temperature. Instead, the box method provides a linear approximation of the error drift over the temperature range. Second, the box method does not identify the temperature at which the minimum and maximum error values occur. Instead, the drift specification just bounds the minimum and maximum values within the temperature range, but they can theoretically occur anywhere within those bounds.

When possible, a better way to identify how an error behaves across temperature is to use one of the datasheet error plots



Shown here are two over-temperature error plots from the ADS124S08 datasheet. The top plot shows the offset voltage in microvolts for each gain from -50 degrees Celsius to 125 degrees Celsius, while the bottom plot shows the gain error in PPM over the same temperature range. Note that while neither plot is linear, they are not quite as arbitrary as the example plot shown on the previous slide. Let's discuss some of the key takeaways from these plots

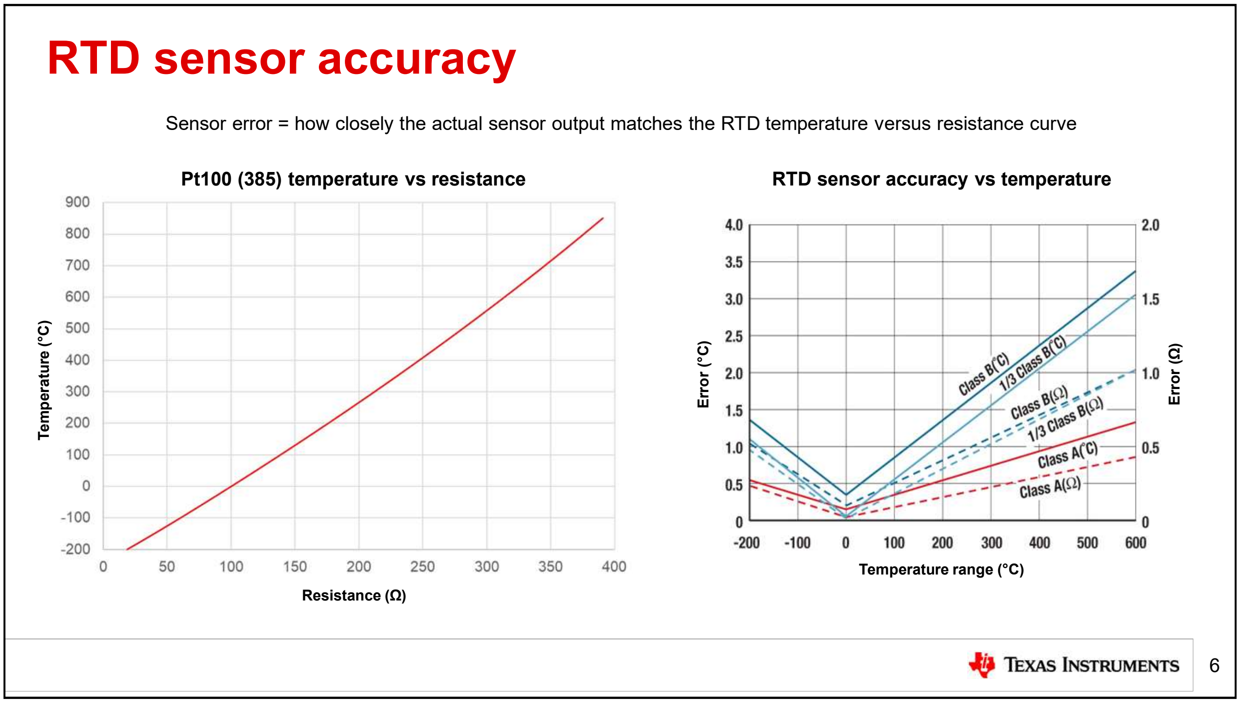
First, the data used to create these plots is only taken at specific temperatures, such as -50, 25, 85, and 125 degrees Celsius as shown. The temperature error is then assumed to be linear between these points. Therefore, the plot is not an accurate representation of the error at *every* ambient temperature, but instead indicates the general behavior across temperature

Next, both plots have a region where the error is approximately linear across temperature. This is highlighted by the orange rectangles. Using the box method to calculate a temperature drift specification within these regions provides a reasonably accurate estimate of the drift. Comparatively, the error increases rapidly at higher temperatures, which is highlighted by the blue boxes. More nonlinearity is common behavior at the temperature extremes. As a result, using a drift specification derived by the box method over the entire temperature range would overestimate the error up to approximately 85 degrees Celsius and underestimate the error beyond this temperature.

One benefit of the over temperature plots is that they provide a more accurate representation of the error, whereas the specified values provide a single linear approximation. The actual response allows the user to better determine if over-temperature calibration is necessary, as well as the specific temperatures at which to calibrate. These topics are discussed in more detail in a subsequent Precision Labs module on calibration

Ultimately, it is best practice to use the datasheet error plots for the most accurate temperature drift information. However, if the over-temperature plots are not included in the datasheet, using the specified values is the next best option. Moreover, with the information presented in this module, it is more clear how these specifications are derived and what they actually reveal about the behavior of an error source across temperature

That concludes our discussion about temperature drift errors. Next, let's consider the different RTD sensor errors

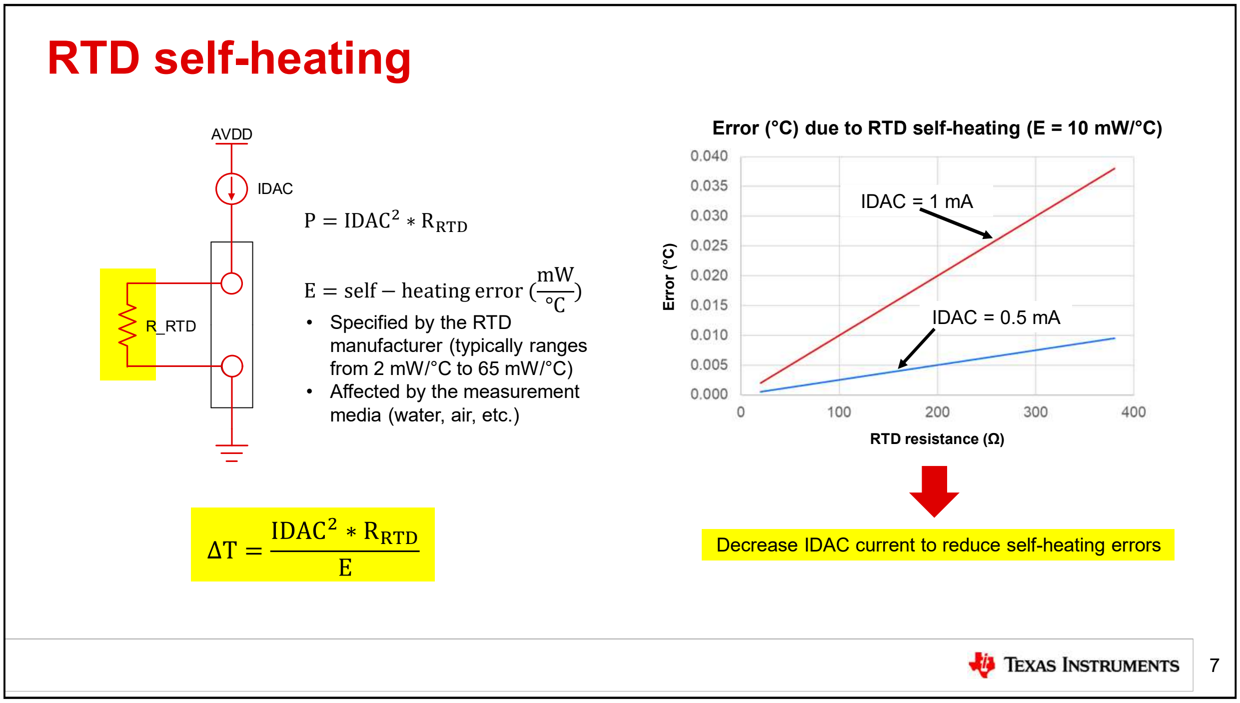


The first RTD error source we need to consider is the variation in the sensor accuracy, which measures how closely the actual sensor output matches the RTD temperature versus resistance curve

For example, a Pt100 resistance versus temperature curve is shown on the left. As noted previously, this curve is derived from a look-up table or calculated using a polynomial and is intended to represent the expected response of a Pt100 RTD. A real RTD however is a physical sensor that has manufacturing tolerances and other minor defects. Therefore, the output response is not perfect

As the plot on the right shows, real RTD sensors have error profiles. All of these profiles share the same general behavior, where the error is approximately 0 at 0 degrees Celsius, then increases linearly for both positive and negative temperatures. However, different classes of RTDs result in different levels of error. For example, a class A RTD has a tighter error specification compared to a class B RTD. Consequently, a class A RTD is more expensive.

RTD sensor accuracy can only be removed by using the actual sensor in the calibration process. However, this also requires a high-accuracy temperature chamber to ensure that the RTD is measuring a known, predictable temperature. This is not practical in most cases, so most RTD measurement systems are specified by the error they contribute without the RTD. Then, it is up to the end-user to decide which class of RTD to pair with their system



Another important RTD error source results from the effects of self heating. As the IDAC current in red flows through the RTD, the RTD produces heat due to power consumption. The power consumed by the RTD is shown as the IDAC current squared multiplied by the RTD resistance

Additionally, the sensor has a specified self-heating error, E, that is usually given in milliwatts per degree Celsius. As noted on the slide, E is a parameter that depends on the sensor construction and is specified by the RTD manufacturer. Typical values range from two to 65 milliwatts per degree Celsius. However, the measurement media also affects how much self-heating an RTD experiences. For example, measuring the temperature of flowing water reduces the effects of RTD self-heating compared to measuring the temperature of still air

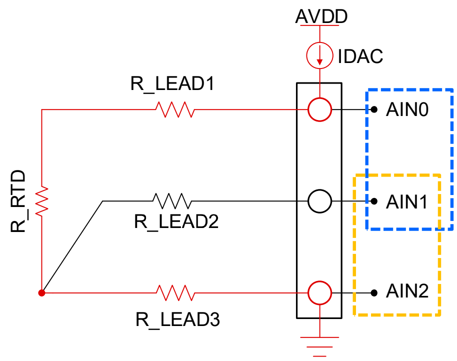
Combining these two terms allows us to determine an equation for the expected temperature error, delta T, which is given on the bottom left. The graph on the right plots this error across RTD resistance for two bias current values. The red plot shows the error when the IDAC current is one milliamp, while the blue plot shows the error when the IDAC current is 0.5 milliamps

The most important takeaway from this plot is that under equal conditions, doubling the IDAC current results in a four-fold increase in the self-heating error due to the power dissipation equation. Therefore, consider using smaller IDAC currents if RTD self-heating is a concern for your system

**Review 3-wire RTD video for more information

RTD lead resistance mismatch (3-wire RTDs only)

3-wire RTD with 1x IDAC**



Input voltage calculations

$$\text{VIN\_final} = \boxed{(\text{V\_AIN0} - \text{V\_AIN1})} - \boxed{(\text{V\_AIN1} - \text{V\_AIN2})}$$

$$\text{VIN}_{\text{Mismatch}} = \text{IDAC} * (\text{R\_RTD} + \text{R\_LEAD1} - \text{R\_LEAD3})$$

$$\text{VIN}_{\text{No mismatch}} = \text{IDAC} * \text{R\_RTD}$$

Error calculations

$$\text{Error} = \left(\frac{\text{VIN}_{\text{Mismatch}} - \text{VIN}_{\text{No mismatch}}}{\text{VIN}_{\text{No mismatch}}}\right) * 100\%$$

$$\text{Error} = \left(\frac{\cancel{\text{IDAC} *} (\cancel{\text{R\_RTD}} + \text{R\_LEAD1} - \text{R\_LEAD3}) - \cancel{\text{IDAC} * \text{R\_RTD}}}{\cancel{\text{IDAC}} * \text{R\_RTD}}\right) * 100\%$$

$$\text{Error} = \left(\frac{\text{R\_LEAD1} - \text{R\_LEAD3})}{\text{R\_RTD}}\right) * 100\%$$

Keep lead resistance small compared to R_RTD

TEXAS INSTRUMENTS 8

The last sensor error we need to consider is lead resistance mismatch error. However, it is important to note that this type of error only affects 3-wire RTDs because of how the lead resistance cancellation is performed. Comparatively, 2-wire RTDs do not offer lead resistance cancellation, and 4-wire RTDs do not include the lead resistance as part of the measurement

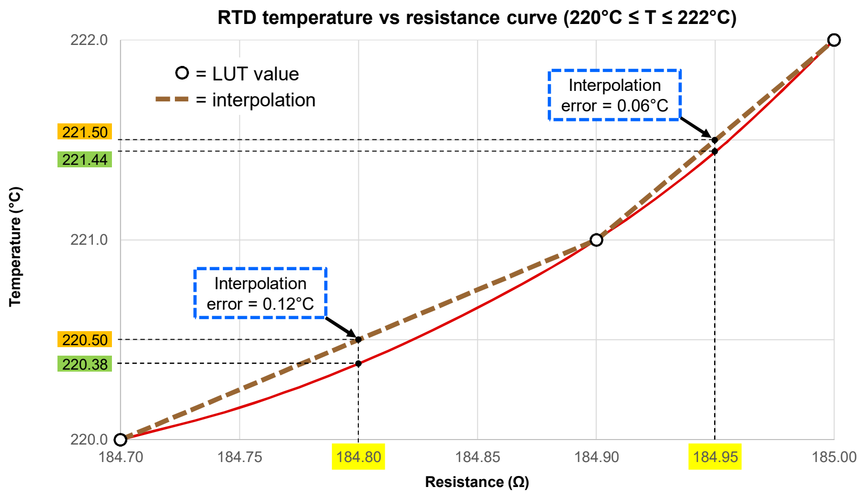
As an example, a 3-wire RTD using one IDAC is shown on the left. Note that this same analysis applies to 3-wire RTDs using two IDACs, though the resulting equations might be slightly different. Moreover, 3-wire RTD system using two IDACs require consideration of IDAC mismatch, which is not discussed in this presentation. Instead, review the Precision Labs module on challenges with 3-wire RTD systems for more information about how these circuits behave and how any equations are derived. The calculated input voltage for this system is shown on the top right as V_IN_final. This specific 3-wire RTD circuit requires two measurements to implement lead resistance cancellation, which are highlighted in blue and orange

Next, we can substitute the IDAC, R_RTD, and R_LEAD terms into the input voltage formula to derive equations for a system with and without lead resistance mismatch. These equations are shown in the upper right. Note that the No Mismatch equation is the same as the Mismatch equation, but the No Mismatch equation assumes R_LEAD1 equals R_LEAD3. Now we are able to calculate the percent error as a result of lead resistance mismatch

First, let's start with the general form of the error equation shown on the bottom right. We take the difference between the ideal and mismatched voltages, then divide this result by the ideal voltage to calculate the error in percent. Inputting the V_IN terms from the input voltage calculations section yields the second equation on the bottom right

Finally, we can divide out the IDAC term and cancel R_RTD in the numerator. The resulting error equation, highlighted in yellow, states that the error in percent is the difference between the lead resistances divided by the RTD resistance. Therefore, the error will be more significant as the RTD resistance decreases or as the lead resistance increases. While a change in RTD resistance depends on the RTD type and measurement range, lead wire resistance increases as the lead wire length increases or as the cross-sectional area decreases. Ensure that the wiring is sized appropriately to minimize lead resistance mismatch in your system

Resistance-to-temperature conversion errors



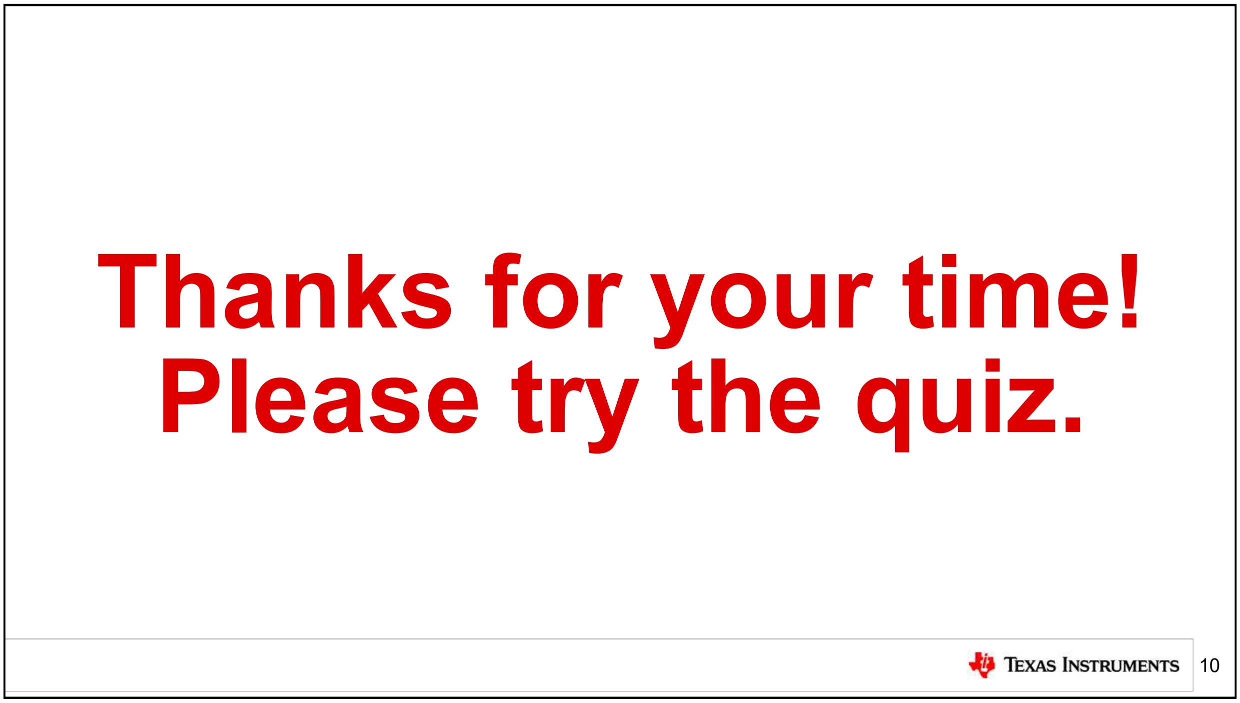
Finally, the resistance to temperature conversion process can also be a source of error in RTD measurement systems. A previous Precision Labs module on RTD measurement circuits introduced the concept of linear interpolation, which is a simple method to calculate a temperature from a measured resistance that is not directly included in the look-up table. As its name implies, linear interpolation assumes that the actual RTD response is linear between the points given in the look-up table. However, since an RTD response curve such as the one shown here is derived from a polynomial, linear interpolation errors can occur

For example, let's assume the system measures an RTD whose resistance at some unknown temperature is 184.8 ohms, which is highlighted in yellow. Assuming the precise equation for this RTD curve is known, the controller calculates that this resistance value corresponds to a measured temperature of 220.38 degrees Celsius. This value is highlighted in green

Now, recall that an RTD look-up table is comprised of a discrete set of points from the RTD curve. Typically, the temperature values in an RTD look-up table are given as integers in one degree Celsius increments, which in this example are 220, 221, and 222 degrees Celsius. Any measured temperature that falls between these points requires the linear interpolation algorithm. In this specific case, the linear interpolation is performed between 220 and 221 degrees Celsius as shown by the brown, dashed line. Applying the interpolation algorithm results in a calculated temperature of 220.5 degrees Celsius, highlighted in orange. This represents an error of 0.12 degrees Celsius compared to the actual temperature

A similar interpolation can be performed between 221 and 222 degrees Celsius, where a measured resistance of 184.95 ohms yields a 0.06 degree Celsius difference between the actual temperature and the interpolated value. Both of these results illustrate that a look-up table and the linear interpolation algorithm make assumptions about the shape of the actual RTD response, and those assumptions can lead to calculation errors

Using polynomial equations instead of look-up tables can help avoid linear interpolation errors because a polynomial equation is better at representing the actual shape of the RTD response curve. However, as noted previously, polynomials require more complex math and therefore more computational power. Additionally, complex math in a processor can introduce under- or over-flow conditions as well as precision errors due to the choice of fixed versus floating point numbers. These types of errors are not specific to RTDs, and as such are not discussed further in this Precision Labs module series.



That concludes this video. Thank you for watching. Please try the quiz to check your understanding of this video's content.

Quiz: over-temperature & additional error sources

1. What is the disadvantage of using polynomials to convert RTD resistance to temperature?
   a. They require more complex math and therefore more computational power
   b. They are not as accurate as a look-up table
   c. They only work over a limited temperature range

2. What does the RTD class define?
   a. The RTD temperature range
   b. The RTD moisture sensitivity
   c. The RTD robustness versus mechanical shock
   d. The RTD accuracy

Question 1. What is the disadvantage of using polynomials to convert RTD resistance to temperature?

The correct answer is A, they require more complex math and therefore more computational power

Question 2. What does the RTD class define?

The correct answer is D, the RTD accuracy.