NAME: AMOO OLUWAFERANMI MARK

MATRIC NO. : 18/ENG04/077

DEPARTMENT : ELECTRICAL AND ELECTRONICS ENGINEERING

LEVEL : 400 L

COURSE CODE : EEE 471

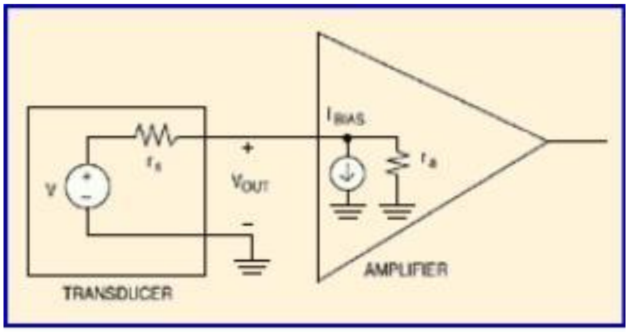
Course Title : Electronic Instrumentation(Digital)

ASSIGNMENT

1. Explain briefly the signal processing and interfacing techniques in measuring instruments

   Answer
   Many sensors provide a voltage output in response to a detected phenomenon. Most of these devices, though, don't produce signals strong enough to be fed directly into an A/D converter or to drive indicators or actuators. To remedy this, analog signal processing often has to boost the sensor's output.

   A. Thevanin-Equivalent Model
   Voltage-output transducers measure a variety of effects, but their electrical interfaces can be roughly modeled by an ideal voltage source in series with an impedance (see Figure 1). Even though the impedance can result from capacitance or inductance (and vary with signal frequency), it can still be considered a simple resistance for many back-of-envelope calculations. Impedance limits the energy the signal source can deliver to a load. The same effect prevents you from starting your car's engine with 8 AA batteries in series



$$V_{OUT} = V_{OUT}\left(\frac{R_0}{R_0 + R_S}\right)$$

(1.5 V each), even though the total voltage is the same as that of an automobile battery. The difference is that the equivalent resistance of the AA batteries is several orders of magnitude higher than that of the car battery, limiting the peak current they can deliver to a few amperes, as opposed to the several hundred amperes provided by a car battery.

Although few sensors must deliver amperes of current, many have high output impedances (i.e., >106 Ω) and are easy to improperly load. In the

Figure 2. The inputs to amplifier stages can also be modeled as Thevanin-equivalent circuits. You generally want to use an amplifier with an input impedance much higher than the sensor's output impedance to reduce system gain errors.

case of a sensor output, a Thevanin equivalent circuit can describe the input to an amplifier or other interface circuit.

Figure 2 shows a sensor and an amplifier. The addition of the amplifier has two effects on the output of the sensor. First, the input impedance of the amplifier forms a voltage divider with the output impedance of the sensor, reducing the sensor's output voltage by:

The second effect is caused by the amplifier's bias current (input current source). Although this current is often in the nanoamp or picoamp range, it can cause input offset voltage errors of several millivolts for a sensor with sufficiently high output impedance (Ibias x Rs).

The lesson to be learned here is to select amplifiers with impedances much higher than those of the sensors you are going to use. Also, make sure the amplifiers have bias currents low enough to avoid creating unacceptable input offset errors.

Although the notion of matching input and output impedances is common in the RF, video, and audio worlds, it's usually neither necessary nor desirable for low-frequency signals (i.e., <1 kHz) that are carried over distances of a few meters or less. In most cases, the DC gain and offset errors caused by attempting to match an amplifier input impedance with a sensor output impedance (which can be highly variable anyhow) will far exceed the AC errors caused by mismatched impedances among sensor, interconnecting cable, and a high-impedance amplifier.

B. Single-Ended vs. Differential Measurements

A single voltage sensor output with respect to ground is called a single-ended output; transducers that provide two outputs, where the second either remains constant or changes with an opposite polarity to the first, produce differential or balanced outputs. Single-ended outputs have the advantage of simplicity, but they are more susceptible to interference and signal degradation than differential outputs. In cases where the sensor signal is small and rides on a significant DC bias, a balanced output lets you more easily discriminate changes, especially when the DC bias changes in response to environmental factors, such as temperature.

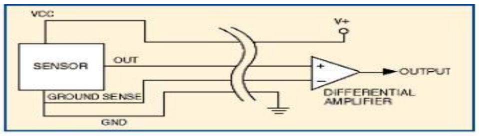
Figure 3 shows two thermistor-based temperature-sensing schemes; one is single-ended, and the other is differential. The differential scheme allows for an order of magnitude or more of sensitivity with the same voltmeter because most of the voltmeter's dynamic range isn't consumed in measuring the bias voltage level, as it is in a single-ended measurement.

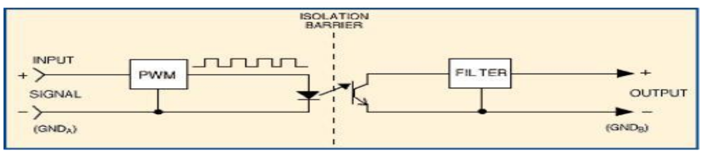
Because differential measurement schemes are popular, single-IC instrumentation amplifiers (see "Instrumentation Amplifiers: A Tutorial," Sensors, September 1997) are available to amplify differential signals and convert them to single-ended signals for subsequent processing.

C. Ground and Isolation

Ground is the point at which the voltage is taken to be zero. Unfortunately, voltage levels at one ground are not always the same as they are at another, and this is where the problems begin. Ground variations (measured at the wall outlet) of a few tens of millivolts AC (60 Hz) are not uncommon in the same building. Such small variations don't often present major safety issues, but they can make remote voltage measurements difficult.

One approach is to make sure that the sensor being measured is grounded only at the amplifier inputs. For passive sensors (i.e., those requiring no external power to operate), the signal lead and the return can be brought straight to the amplifier. For active sensors, the technique is more difficult to use because voltage drops along the return lead can lift the ground at the sensor millivolts or even volts above the ground at the amplifier inputs.





You can avoid the problem by using a differential output sensor, which will cancel out a few hundred millivolts of sensor ground error. For single-ended

Figure 3. Remote sensing lets you avoid offset errors caused by ground shifts on long cable runs. By subtracting the ground potential at the sensor from the sensor's output signal, you can obtain a true output value.

sensors, a technique called remote ground sensing can be used (see Figure 3). By measuring the ground at the sensor and subtracting that value from the sensor output signal (using an instrumentation amplifier), you can obtain the true sensor output. This technique works because there is no voltage drop along wires that don't carry current, which is the case for the sensor output and ground sense leads if the instrumentation amplifier has sufficiently high input impedance and low input bias current.

Sometimes the difference in ground from the beginning of the measurement chain to its end is tens or hundreds of volts. In cases like these, simple differential or ground-sensing schemes won't yield effective results, and in many cases, using these schemes can damage the measuring equipment or injure the people operating it.

D. Low-Voltage Signals

Transducers often output microvolt signals, and you encounter difficulties when you try to accurately measure such small signals. The major difficulties are intrinsic noise from the sensor and the amplifier, thermal errors, and EMI.

Electronic devices produce electrical noise, and these noise sources set the lower bound on signal recovery. One such source is Johnson noise, which is generated by resistors. The noise is

not dependent on the resistor type or construction, only on the resistor value in ohms and the resistor's temperature. Johnson noise voltage (RMS) for a resistor of R ohms is given by:
where:

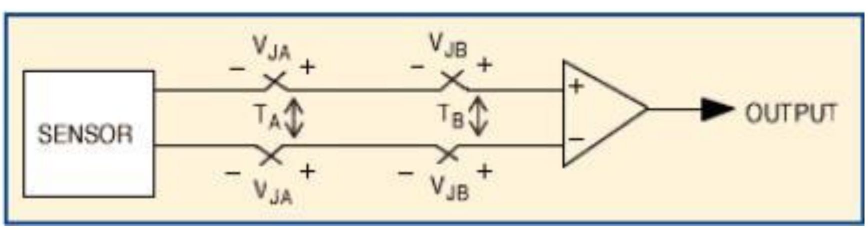
k = Boltzman's constant (1.38 x 10-23 J/°K) T = temperature in °K

B = bandwidth in Hz

At room temperature (293ºK), a 1 KΩ resistor generates 0.4 µV of noise over a 10 kHz bandwidth. The Johnson noise developed by a transducer's output resistance sets the lower limit on the recoverable signal. In many cases, however, the noise sources peculiar to a particular type or model of transducer can be an order of magnitude greater than the Johnson noise.

Solid-state amplifiers also contribute noise to the signal processing chain. Although several effects contribute to the noise performance of an amplifier, amplifiers can be well characterized by an equivalent input voltage noise and an equivalent input current noise (in), both of which vary over frequency (see Figure 4).

$$V_n = \sqrt{e_n^2 + \left(i_n \bullet \frac{R_s \bullet R_i}{R_s + R_i}\right)^2}$$



Voltage noise is specified as if it were a voltage source placed in series with the input of a noiseless amplifier. Current noise is specified as a current source in parallel with the amplifier input and converted to voltage noise by flowing through the output impedance of the transducer in parallel with the amplifier's input impedance. Because the two noise sources are uncorrelated, their voltages do not simply add; an expression for total input noise is given by: Because the voltage and current noise sources are independent of each other, the choice of amplifier technology (Bipolar or FET) is dependent on the source impedance of the transducer.

Bipolar amplifiers tend to have lower input voltage noise, and FET amplifiers have lower input current noise, at least when considering top-grade devices. Thus, for low-impedance transducers, a bipolar front-end amplifier may be the best choice because voltage noise will be higher than current noise, and for high-impedance sources, an FET front end may be the better choice because input current noise will be the most troublesome.

A popular temperature sensor, the thermocouple, consists of a junction of two different metals, which develops a temperature-dependent voltage. The thermoelectric voltage is typically quite small (microvolts) and can be a challenge to measure accurately.

A similar but opposite situation exists when you try to measure microvolt signals. Every bimetallic junction in a measurement circuit (e.g., solder joints and wire connections) contributes a small, temperature-dependent error voltage. The challenge then is to be able to ignore the effects of the parasitic thermocouple signals so that you can accurately measure the signal of interest.

2. Explain briefly the expert system instrumentation

In artificial intelligence, an expert system is a computer system emulating the decision-making ability of a human expert.Expert systems are designed to solve complex problems by reasoning through bodies of knowledge, represented mainly as if–then rules rather than through conventional procedural code. The first expert systems were created in the 1970s and then proliferated in the 1980s.Expert systems were among the first truly successful forms of artificial intelligence (AI) software An expert system is divided into two subsystems: the inference engine and the knowledge base. The knowledge base represents facts and rules. The inference engine applies the rules to the known facts to deduce new facts. Inference engines can also include explanation and debugging abilities.

An expert system is an example of a knowledge-based system. Expert systems were the first commercial systems to use a knowledge-based architecture. A knowledge-based system is essentially composed of two sub-systems: the knowledge base and the inference engine.

The knowledge base represents facts about the world. In early expert systems such as Mycin and Dendral, these facts were represented mainly as flat assertions about variables. In later expert systems developed with commercial shells, the knowledge base took on more structure and used concepts from object-oriented programming. The world was represented as classes, subclasses, and instances and assertions were replaced by values of object instances. The rules worked by querying and asserting values of the objects.

The inference engine is an automated reasoning system that evaluates the current state of the knowledge-base, applies relevant rules, and then asserts new knowledge into the knowledge base. The inference engine may also include abilities for explanation, so that it can explain to a user the chain of reasoning used to arrive at a particular conclusion by tracing back over the firing of rules that resulted in the assertion.

There are mainly two modes for an inference engine: forward chaining and backward chaining. The different approaches are dictated by whether the inference engine is being driven by the antecedent (left hand side) or the consequent (right hand side) of the rule. In forward chaining an antecedent fires and asserts the consequent. For example, consider the following rule:

R 1 : M a n (x) ⟹ M o r t a l (x) {\displaystyle R1:{\mathit {Man}}(x)\implies {\mathit {Mortal}}(x)}

{\displaystyle R1:{\mathit {Man}}(x)\implies {\mathit {Mortal}}(x)}

A simple example of forward chaining would be to assert Man(Socrates) to the system and then trigger the inference engine. It would match R1 and assert Mortal(Socrates) into the knowledge base. Backward chaining is a bit less straight forward. In backward chaining the system looks at possible conclusions and works backward to see if they might be true. So if the system was trying to determine if Mortal(Socrates) is true it would find R1 and query the knowledge base to see if Man(Socrates) is true. One of the early innovations of expert systems shells was to integrate inference engines with a user interface. This could be especially powerful with backward chaining. If the system needs to know a particular fact but does not, then it can simply generate an input screen and ask the user if the information is known. So in this example, it could use R1 to ask the user if Socrates was a Man and then use that new information accordingly. The use of rules to explicitly represent knowledge also enabled explanation abilities. In the simple example above if the system had used R1 to assert that Socrates was Mortal and a user wished to understand why Socrates was mortal they could query the system and the system would look back at the rules which fired to cause the assertion and present those rules to the user as an explanation. In English, if the user asked "Why is Socrates Mortal?" the system would reply "Because all men are mortal and Socrates is a man".

As expert systems evolved, many new techniques were incorporated into various types of inference engines. Some of the most important of these were:

- Truth maintenance. These systems record the dependencies in a knowledge-base so that when facts are altered, dependent knowledge can be altered accordingly. For example, if the system learns that Socrates is no longer known to be a man it will revoke the assertion that Socrates is mortal.
- Hypothetical reasoning. In this, the knowledge base can be divided up into many possible views, a.k.a. worlds. This allows the inference engine to explore multiple possibilities in parallel. For example, the system may want to explore the consequences of both assertions, what will be true if Socrates is a Man and what will be true if he is not?
- Uncertainty systems. One of the first extensions of simply using rules to represent knowledge was also to associate a probability with each rule. So, not to assert that

Socrates is mortal, but to assert Socrates *may* be mortal with some probability value. Simple probabilities were extended in some systems with sophisticated mechanisms for uncertain reasoning, such as Fuzzy logic, and combination of probabilities.

- Ontology classification. With the addition of object classes to the knowledge base, a new type of reasoning was possible. Along with reasoning simply about object values, the system could also reason about object structures. In this simple example, Man can represent an object class and R1 can be redefined as a rule that defines the class of all men. These types of special purpose inference engines are termed classifiers. Although they were not highly used in expert systems, classifiers are very powerful for unstructured volatile domains, and are a key technology for the Internet and the emerging