EXERCISE NO.: 4

QUADRATURE DECODING FOR POSITION SENSING

DATE: 18-09-2020

Reg. No. : RA1711018010101

LAB PREREQUISITES:

None

PREREQUISITE KNOWLEDGE:

Fundamentals of MATLAB Simulink model.

OBJECTIVES:

➢ To understand the various encoding types in optical incremental encoders (quadrature encoders)

➢ To develop a Simulink model for sensing position from quadrature encoders

THEORY

Encoders:

An encoder is an electromechanical device that can measure motion or position. Most encoders use optical sensors to provide electrical signals in the form of pulse trains, which can, in turn, be translated into motion, direction, or position.

Rotary encoders are used to measure the rotational motion of a shaft. The following figure shows the fundamental components of a rotary encoder, which consists of a light-emitting diode (LED), a disk, and a light detector on the opposite side of the disk. The disk, which is mounted on the rotating shaft, has patterns of opaque and transparent sectors coded into the disk. As the disk rotates, the opaque segments block the light and, where the glass is clear, light is allowed to pass. This generates square-wave pulses, which can then be interpreted into position or motion.
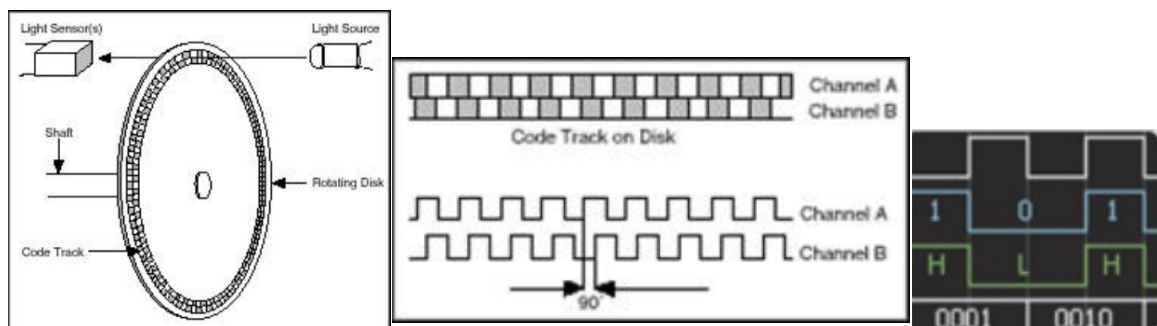
Encoders usually have from 100 to 6,000 segments per revolution. This means that these encoders can provide 3.6 deg of resolution for the encoder with 100 segments and 0.06 deg of resolution for the encoder with 6,000 segments. Linear encoders work under the same principle as rotary encoders except that instead of a rotating disk, there is a stationary opaque strip with transparent slits along its surface, and the LED- detector assembly is attached to the moving body.

An encoder with one set of pulses would not be useful because it could not indicate the direction of rotation. Using two code tracks with sectors positioned 90 deg out of phase, the two output channels of the quadrature encoder indicate both position and direction of rotation. If A leads B, for example, the disk is rotating in a clockwise direction. If B leads A, then the disk is rotating in a counter-clockwise direction. Therefore, by monitoring both the number of pulses and the relative phase of signals A and B, you can track both the position and direction of rotation.

In addition, some quadrature encoders include a third output channel – called a zero or reference signal – which supplies a single pulse per revolution. You can use this single pulse for precise determination of a reference position. In the majority of encoders, this signal is called the Z-Terminal or the index.

So far, this document has addressed only what are called single-ended incremental quadrature encoders. These are called single-ended because the A and B signals are both referenced to ground, so there is one wire (or end) per signal. Another commonly used type of encoder is a differential encoder, where there are two lines per each A and B signal. The two lines for the A signal are A' and A, and the two lines for the B signal are B' and B. This type of configuration is also called push-pull because all four lines are always supplying a known voltage (either 0 V of Vcc). When A is Vcc, A' is 0 V, and when A is 0 V, A' is Vcc. In the case of a single-ended encoder, A is either Vcc or it floats. Differential encoders are often used in electrically noisy environments because taking differential measurements protects the integrity of the signal.

To make encoder measurements, you need a basic electronic component called a *counter*. Based on its several inputs, a basic counter emits a value that represents the number of edges (low to high transitions in the waveform) counted. Most counters have three relevant inputs – *gate, source,* and *up/down.* The counter counts the events registered in the *source* input, and, depending on the state of the *up/down* line, it either increments the count or decrements it. For example, if the *up/down* line is "high" the counter increments the count, and if it is "low," the counter decrements the count. An encoder usually has five wires that you need to connect to the instrument, and, depending on the encoder, these wires vary in color. You can use these wires to provide power to the encoder and to read in the A, B, and Z signals. Figure 4 shows a typical pinout table for an incremental encoder.

Pin	5-pin Single-ended
1	Ground
2	Index
3	A channel
4	+5VDC power
5	B channel

The next step is determining where you should connect each of these wires. Considering the counter described above, signal A is connected to the source terminal, making this the signal from which the pulses are counted. Signal B is connected to the up/down terminal, and you can connect the +5 VDC and ground signals to any power source – in most cases, a digital line in a data acquisition device card suffices.

Once the edges are counted, the next concept you need to consider is how those values are converted to position. The process by which edge counts are converted to position depends on the type of encoding used. There are three basic types of encoding, X1, X2, and X4.
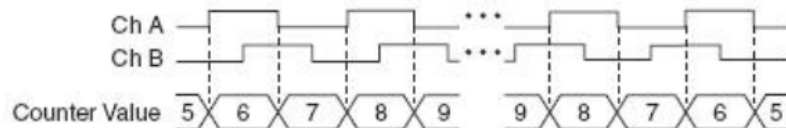
X1 Encoding

The following figure shows a quadrature cycle and the resulting increments and decrements for X1 encoding. When channel A leads channel B, the increment occurs on the rising edge of channel A. When channel B leads channel A, the decrement occurs on the falling edge of channel A.
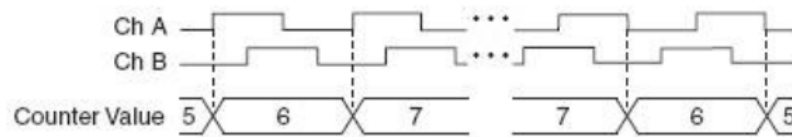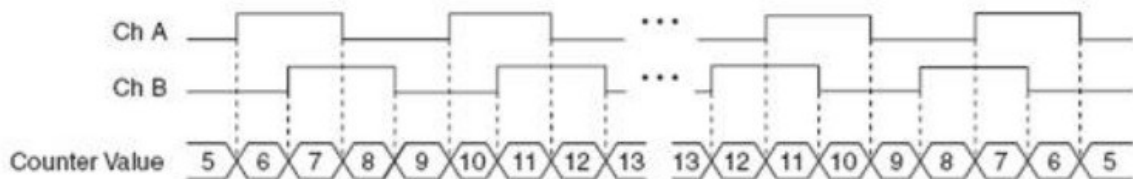

X2 Encoding

The same behavior holds for X2 encoding except the counter increments or decrements on each edge of channel A, depending on which channel leads the other. Each cycle results in two increments or decrements, as shown in figure below.

X4 Encoding

The counter increments or decrements similarly on each edge of channels A and B for X4 encoding. Whether the counter increments or decrements depends on which channel leads the other. Each cycle results in four increments or decrements, as shown in the figure below.

Once you have set the encoding type and counted the pulses, converting to position is a matter of using one of the following formulas:

For Rotational Position

Amount of Rotation in radians = $\dfrac{\text{Total count} * 2*\text{pi}}{N*x}$

where N = number of pulses generated by the encoder per shaft revolution

x = encoding type (x=1 for X1 decoding; x=2 for X2 decoding; x=4 for X4 decoding)

EXERCISE TASKS

Model of a DC Motor

To add actuator to the actuated link model, use the following parameters.

Physical Parameter	Specification
Armature resistance	1 Ohm
Electrical inductance	0.5 H
Torque Constant(K)	0.01 Nm/A (or) V/(rad/s)
Damping	0.1 Nm/(rad/sec)
Moment of Inertia	0.01 kg·m^2
Ideal torque source (Load)	0.001 Nm
Maximum Voltage	±20V

Model of an Incremental Shaft Encoder

Physical Parameter	Specification
Pulse voltage	1 V
No. of pulses per revolution	20

Task 1: Angular Position in X1 Encoding

- Develop a Simulink model to obtain the angular position of DC motor shaft using the X1 encoding. And also give real angular position of the shaft ranging from 0 to 2*pi rad.
- Use the following logic for decoding
  - Increment the count on a raising edge of A if A leads B
  - Decrement the count on a falling edge of A if B leads A

Task 2: Angular Position in X2 Encoding

- Develop a Simulink model to obtain the angular position DC motor shaft using the X2 encoding. And also give real angular position of the shaft ranging from 0 to 2*pi rad.

- Use the following logic for decoding
  - On raising or falling edge of A:
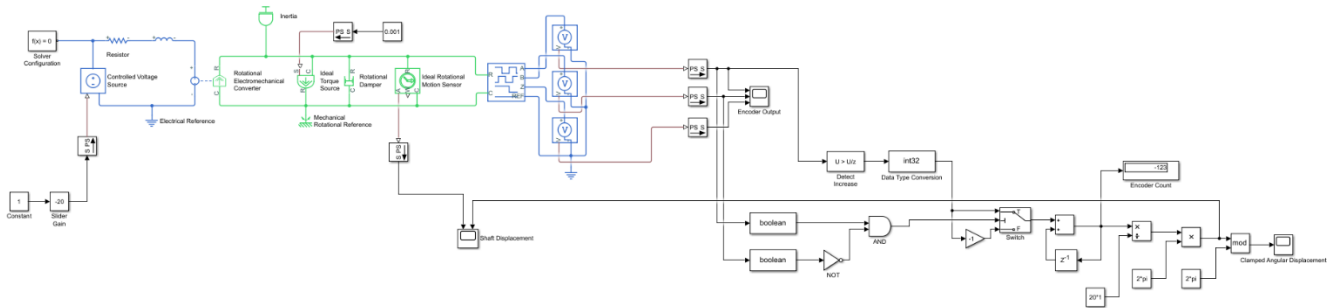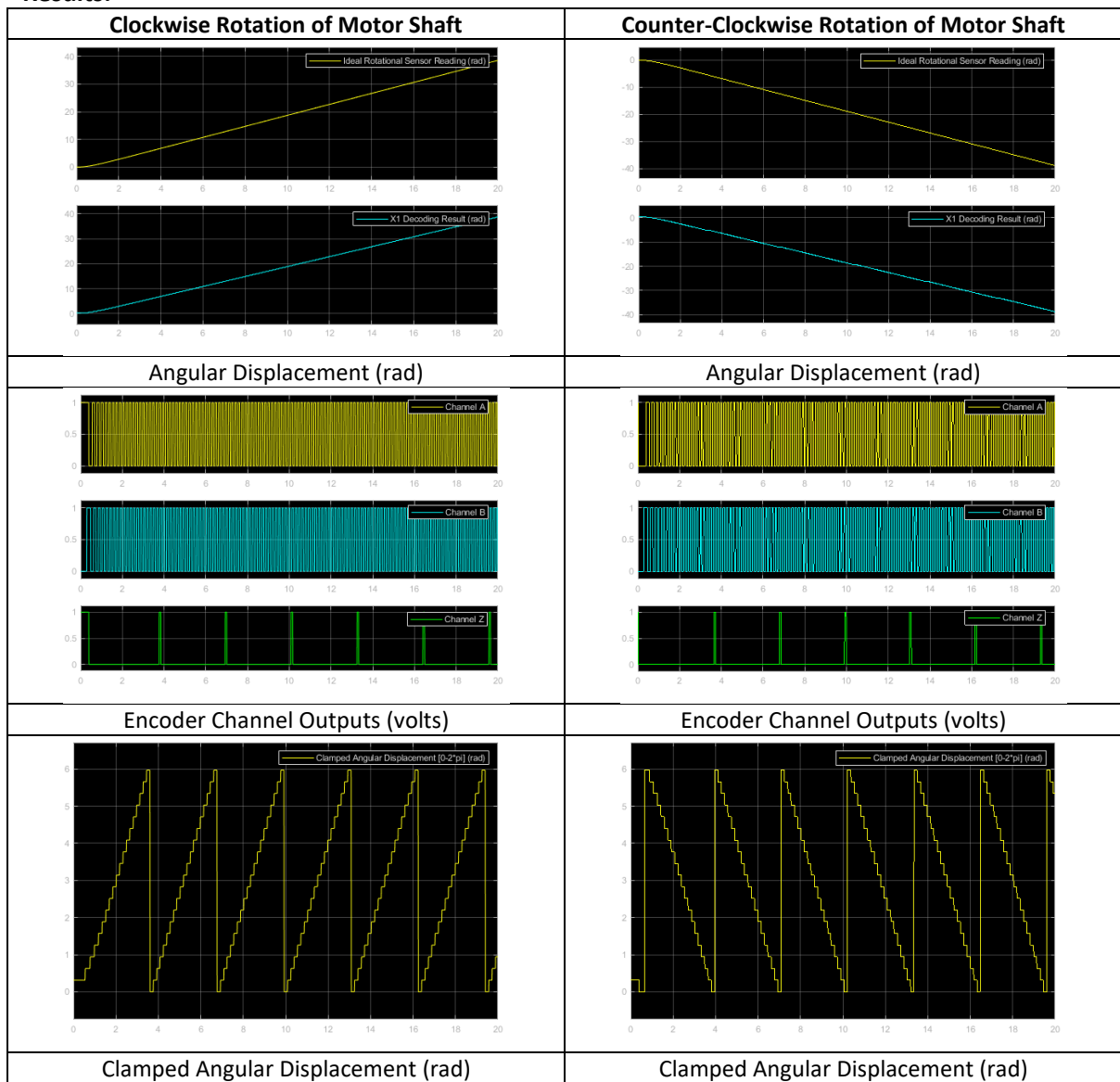   - Increment if A leads B
    - Decrement if B leads A

Task 3: Angular Position in X4 Encoding

- Develop a Simulink model to obtain the angular position of DC motor shaft using the X4 encoding. And also give real angular position of the shaft ranging from 0 to 2*pi rad.
- Use the following logic for decoding
  - On rising or falling edge of A or B
    - Increment if A leads B
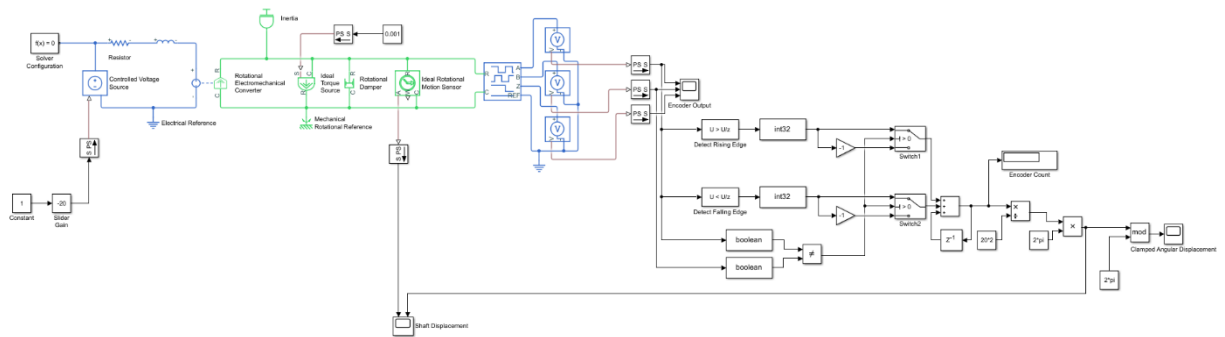   - Decrement if B leads A

Task 4: Compare the Angular Positions Obtained

- Compare the angular positions obtained using the three different encoding types.

	X1 Encoding Mode	X2 Encoding Mode	X4 Encoding Mode
Clockwise direction (positive count)			
Voltage applied to motor in Volts	20	20	20
Run Time in Seconds	20	20	20
Total count	123	246	492
Actual position from built-in sensor in rad	38.6	38.6	38.6
Reading from decoding logic in rad	38.64	38.64	38.64
Counter Clockwise Direction (negative count)			
Voltage applied to motor in Volts	-20	-20	-20
Run Time in Seconds	20	20	20
Total count	-123	-248	-496
Actual position from built-in sensor in rad	-39	-39	-39
Reading from decoding logic in rad	-38.64	-38.96	-38.96

TASK 1:

Block Diagram:

Results:
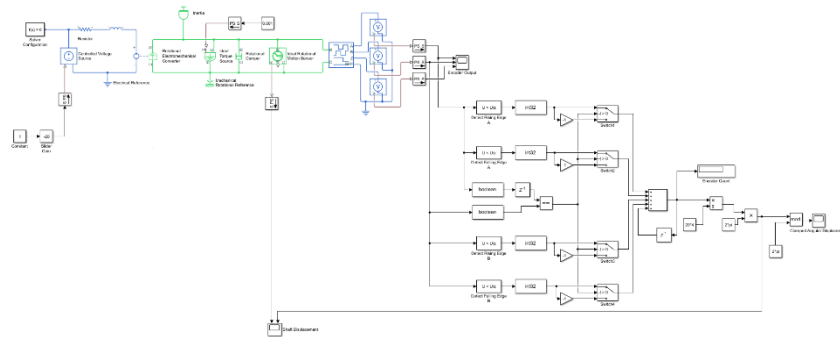
Clockwise Rotation of Motor Shaft	Counter-Clockwise Rotation of Motor Shaft
Angular Displacement (rad)	Angular Displacement (rad)
Encoder Channel Outputs (volts)	Encoder Channel Outputs (volts)
Clamped Angular Displacement (rad)	Clamped Angular Displacement (rad)

TASK 2:

Block Diagram:

Results:

Clockwise Rotation of Motor Shaft	Counter-Clockwise Rotation of Motor Shaft
Angular Displacement (rad)	Angular Displacement (rad)
Encoder Channel Outputs (volts)	Encoder Channel Outputs (volts)
Clamped Angular Displacement (rad)	Clamped Angular Displacement (rad)

TASK 3:

Block Diagram:

Results:

| Clockwise Rotation of Motor Shaft	Counter-Clockwise Rotation of Motor Shaft
Angular Displacement (rad)	Angular Displacement (rad)
Encoder Channel Outputs (volts)	Encoder Channel Outputs (volts)
Clamped Angular Displacement (rad)	Clamped Angular Displacement (rad)

TASK 4:

Angular positions of the motor shaft obtained using the three different decoding techniques (X1, X2 and X4 decoding) were almost same since the entire system was simulated with idealistic conditions and hence there were no losses/errors while detecting and counting rising and/or falling edges of channel A and/or channel B.

However, the resolution of counting angular displacement varied (increasing from X1 to X4 decoding techniques).

The following table summarizes the results:

	X1 Encoding Mode	X2 Encoding Mode	X4 Encoding Mode
Clockwise direction (positive count)			
Total count	123	246	492
Reading from decoding logic in rad	38.64	38.64	38.64
Counter Clockwise Direction (negative count)			
Total count	-123	-248	-496
Reading from decoding logic in rad	-38.64	-38.96	-38.96

LAB SESSION SCREENSHOT:
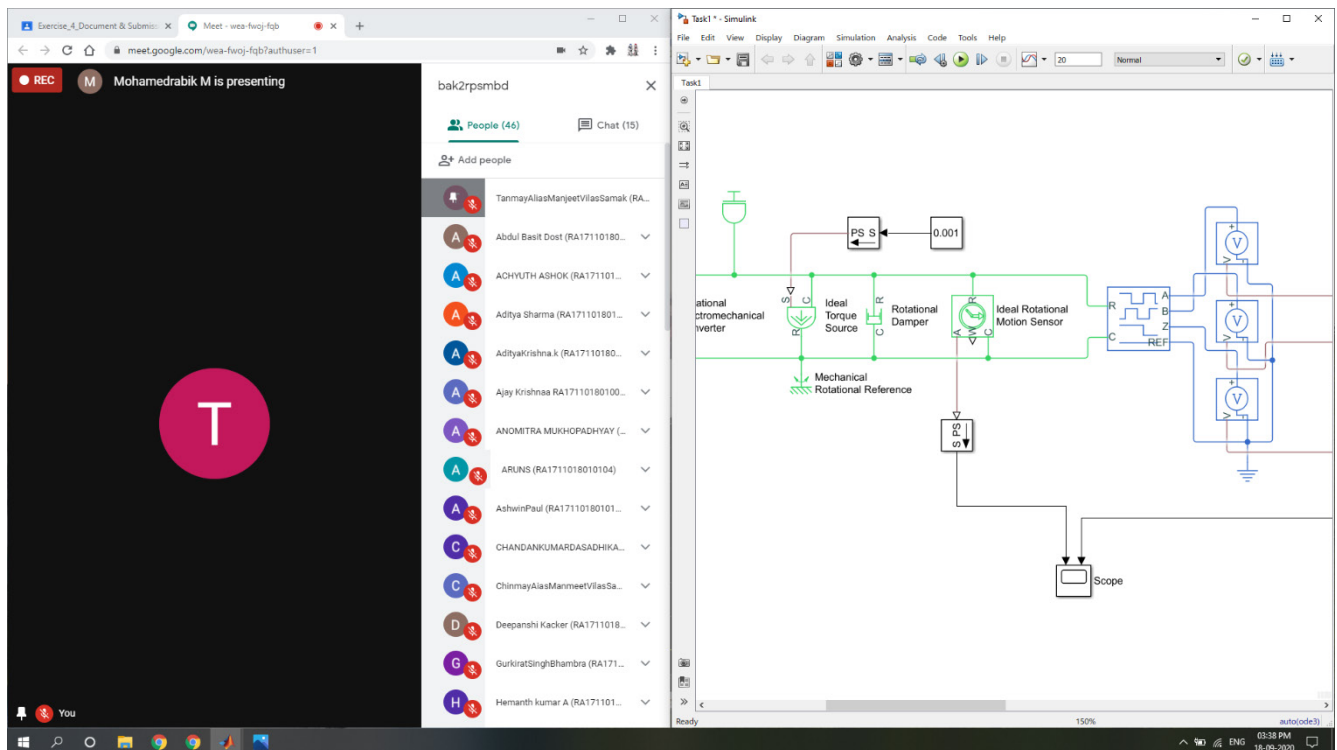
INFERENCE:

- X1 Decoding:
    - Count rising edges of channel A.
    - Up count if $A \cdot \bar{B}$, else down count.
- X2 Decoding:
    - Count rising and falling edges of channel A.
    - Up count if $A \neq B$, else down count.
- X4 Decoding:
    - Count rising and falling edges of channel A and channel B.
    - Up count if $B_t = A_{t-1}$, else down count.

This experiment gave a deeper understanding about optical incremental shaft encoders, and more importantly, the decoding techniques involved to interpret useful information regarding angular displacement of the motor shaft from a train of voltage pulses for each channel obtained as raw data. The various tasks (Task 1-3) in this exercise helped gain a step by step knowledge about modeling and simulation of the entire system starting with X1 decoding and then proceeding with X2 and X4 decoding techniques with self-exploration. Finally, Task-4 helped gain an insight as to how the three decoding techniques are different in terms of the counting resolution.

From this experiment, it is evident that MATLAB - Simulink is a very powerful tool when it comes to modelling and simulation of dynamic systems. It provides a range of built-in functions and toolboxes for rapid system analysis across multiple representation types (including Simulink as well as Physical System blocks – it is an important inference that Simulink blocks represent only numerical data whereas Physical System blocks represent a specific physical quantity along with its associated unit/dimension).